Nonparametric methods for learning and detecting multivariate statistical dissimilarity

Alix Lhéritier

UNIVERSITE NICE SOPHIA ANTIPOLIS

ECOLE DOCTORALE STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

T H E S E

pour obtenir le titre de

Docteur en Sciences

de l'Université Nice Sophia Antipolis

Mention Informatique

présentée et soutenue par

Alix Lhéritier

Nonparametric Methods for Learning and Detecting Multivariate Statistical Dissimilarity

Thèse dirigée par Frédéric CAZALS

soutenue le 23/11/2015

Jury:

Gadiel Seroussi	Professeur, Univ. de la República	Rapporteur
Peter Grünwald	Directeur de Recherche, CWI	Rapporteur
Guillaume Obozinski	Chercheur, École des Ponts ParisTech	Examinateur
Vicente Zarzoso	Professeur, Univ. Nice Sophia Antipolis	Examinateur
Frédéric Cazals	Directeur de Recherche, Inria	Directeur

Contents

Acknowledgments

First of all, I would like to thank my advisor, Dr. Frédéric Cazals, for trusting me and believing in my ideas, for giving me the necessary freedom to work on them and for supporting me with his experience, enthousiasm and positivity all along the process and, especially, in my darkest moments of doubt.

I would like to express my gratitude to Prof. Dr. Peter Grünwald for fruitful discussions and important advice, in particular, with respect to the important optional stopping property of the tests proposed in this thesis, and for allowing me to expand my knowledge in the fascinating field of Statistical Learning. I would also like to thank again my Master's advisors Prof. Dr. Alfredo Viola and Dr. Gadiel Seroussi, for helping me in building my background in Algorithms, Coding and Information Theory, which was essential to this thesis. Additional thanks go to Prof. Dr. P. Grünwald and Dr. G. Seroussi, in their role of "Rapporteur," for their valuable comments and corrections, which greatly improved the quality of this manuscript.

Overall, I am grateful to Inria for providing me excellent conditions to work and, in particular, to cluster administrators for their efficient support when running my multiple experiments. I would also like to thank my labmates Andrea, Angeliki, Deepesh, Dorian, Noël, Romain, Simon and Tom for the discussions and shared fun moments. Special thanks to Tom for his software support.

I would like to thank my parents for supporting and encouraging me during all this time. Special thanks to my mother and Andrzej who made possible and easy our move to the wonderful city of Antibes. In this city, we met many new friends that I would like to thank for their support too. Last but not least, I will never finish to thank my wife Ana and my children Amélie and Yann, for their support and for enduring this long process that sometimes resembled an emotional rollercoaster.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In this thesis, we study problems related to learning and detecting multivariate statistical dissimilarity, which are of paramount importance for many statistical learning methods that are nowadays applied in an increasingly number of fields like biology, medical diagnosis, cheminformatics, fraud detection, computer vision, handwriting recognition, social networks, computational finance, recommendation systems, etc.[1]

Statistical dissimilarity is, in essence, related to the problem of being able to distinguish one random variable from another (i.e., to distinguish their distribution). In particular, we consider the following problems:

- **the *two-sample problem***: given two sets of observations, decide whether they come from the same distribution or not. For example, one may want to find out whether two classes of patients show different response to a drug, where the response can be described by a number of variables. We will also consider the case where these observations can be processed in a sequential manner which allows more flexibility in the sampling process and is of interest in data streaming applications.

- **nonparametric multivariate *effect size* determination**: in both previous cases, we consider unknown multivariate distributions without any assumption on their nature, i.e., the problems are *nonparametric*. Therefore, when a change is detected we are interested in determining the nature of the change. For example, if patients of the different classes show different response to the drug, one may want to identify where this difference is located in the space of variables considered to further infer if it is an intrinsic and important difference or eventually some random error due to the sampling process.

In Section 1.2, we introduce measures of statistical dissimilarity and the related problem of binary classification. In Section 1.3, we show how to learn these objects using data. In Section 1.4, we recall fundamental concepts of hypothesis testing, we present methods for nonparametric two-sample testing.

1.2 Statistical Dissimilarity

In this section, we define some important dissimilarity measures and the related problem of binary classification.

1.2.1 Dissimilarity Measures

Let us first present different ways of quantifying statistical dissimilarity. Let X and Y be two random variables defined on a continuous space Ω with densities f_X and f_Y and cumulative distributions F_X and F_Y respectively.

[1] See, e.g., `https://en.wikipedia.org/wiki/Machine_learning#Applications`

Many useful measures of dissimilarity between distributions $d(F_X, F_Y)$ are not *metrics* but satisfy some of their properties, which are:

1. non-negativity : $d(F_X, F_Y) \geq 0$
2. identity of indiscernibles: $d(F_X, F_Y) = 0 \Leftrightarrow F_X = F_Y \Leftrightarrow f_X = f_Y$ a.e.
3. symmetry: $d(F_X, F_Y) = d(F_X, F_Y)$
4. triangle inequality : $d(F_X, F_Z) \leq d(F_Z, F_Y) + d(F_Y, F_Z)$.

Several dissimilarity measures have been introduced in the literature and some can be derived as special cases of others. We focus on three of them that are of special relevance for our work. We present their versions for continuous Ω since it is our focus, but notice that they can be defined for more general probability measures.

ɟ-divergences A common class of dissimilarity measures is the one of ɟ-*divergences*. These divergences were introduced and studied in [Csi63, Mor63, AS66] and are also know as Csiszár ɟ-divergences, Csiszár-Morimoto divergences or Ali-Silvey distances. Intuitively, they can be seen as an average, weighted by a function ɟ, of the density ratio $\frac{f_X(\cdot)}{f_Y(\cdot)}$. For a strictly convex function ɟ satisfying $ɟ(1) = 0$, the ɟ-divergence between f_X and f_Y is defined as

$$D_ɟ\left(f_X \| f_Y\right) = \int_\Omega ɟ\left(\frac{f_X(z)}{f_Y(z)}\right) f_Y(z) dz \tag{1.1}$$

or, more generally, for probability measures P_X and P_Y such that P_X is absolutely continuous with respect to P_Y[2] as

$$D_ɟ\left(P_X \| P_Y\right) = \int_\Omega ɟ\left(\frac{dP_X}{dP_Y}\right) dP_Y \tag{1.2}$$

where $\frac{dP_X}{dP_Y}$ is the Radon-Nikodym derivative.[3]

An ɟ-divergence is non-negative and satisfies the identity of indiscernibles, but they do not satisfy the two other properties of a metric.

The popular *Kullback-Leibler (KL) divergence* is an instance of this class, obtained with $ɟ(z) \equiv -\log(z)$ and denoted as $D_{\mathrm{KL}}\left(f_X \| f_Y\right)$. Other important dissimilarity measures are obtained with different functions ɟ (see, e.g., [LV06]).

As we will see in Section 1.4.2, several hypothesis tests are built upon ɟ-divergences.

Jensen-Shannon Divergence The KL divergence has two properties that may not be desired, namely, it is unbounded and it is not symmetric. Denoting the continuous entropy by $H\left(f\right) \equiv \int_\Omega f(z)\log(f(z))dz$ and considering the average density $\bar{f} \equiv \theta_0 f_X + (1-\theta_0) f_Y$ with $\theta_0 \in [0,1]$, the Jensen-Shannon Divergence (JSD) [Lin91] is defined as follows

$$JS_{\theta_0}\left(f_X \| f_Y\right) \equiv H\left(\bar{f}\right) - \theta_0 H\left(f_X\right) - (1-\theta_0) H\left(f_Y\right) \tag{1.3}$$

which can be equivalently expressed as

$$\begin{aligned} JS_{\theta_0}\left(f_X \| f_Y\right) &= \int_\Omega \bar{f}(z)\log\bar{f}(z)dz - \theta_0 \int_\Omega f_X(z)\log f_X(z)dz - (1-\theta_0)\int_\Omega f_Y(z)\log f_Y(z)dz & (1.4)\\ &= \theta_0 \int_\Omega f_X(z)\log\frac{\bar{f}(z)}{f_X(z)}dz + (1-\theta_0)\int_\Omega f_Y(z)\log\frac{\bar{f}(z)}{f_Y(z)}dz & (1.5)\\ &\equiv \theta_0 D_{\mathrm{KL}}\left(f_X \| \bar{f}\right) + (1-\theta_0) D_{\mathrm{KL}}\left(f_Y \| \bar{f}\right) & (1.6) \end{aligned}$$

[2] P_X is absolutely continuous with respect to P_Y if for every measurable set A, $P_Y(A) = 0$ implies $P_X(A) = 0$.

[3] The Radon-Nikodym derivative $\frac{dP_X}{dP_Y}$ is the function f such that, for every measurable set A, $P_X(A) = \int_A f dP_Y$. For continuous random variables, the density is the Radon-Nikodym derivative of the corresponding (induced) measure with respect to the Lebesgue measure.

since, by linearity, $\int_\Omega \bar{f}(z)\log\bar{f}(z)dz = \theta_0 \int_\Omega f_X(z)\log\bar{f}(z)dz + (1-\theta_0)\int_\Omega f_Y(z)\log\bar{f}(z)dz$.

Therefore, the JSD avoids infinite values by averaging the KL divergence of each density with respect to the average density. The JSD is non-negative and satisfies the identity of indiscernibles. When $\theta_0 = 1/2$, the JSD is symmetric, bounded between 0 and 1 and its square root yields a metric [ES03].

Interestingly, as will see in Section 1.2.2 and Chapter 2, the JSD is linked to binary classification.

Maximum Mean Discrepancy Let $\mathcal{F}$ be a class of functions $g : \Omega \to \mathbb{R}$, the *maximum mean discrepancy (MMD)* is defined in [FM53, GBR$^+$12] as

$$\mathrm{MMD}[\mathcal{F}, f_X, f_Y] \equiv \sup_{g\in\mathcal{F}} \left(\mathbb{E}_X\left[g(X)\right] - \mathbb{E}_Y\left[g(Y)\right]\right). \tag{1.7}$$

In words, the dissimilarity is characterized by a function g from a class $\mathcal{F}$, whose expected value is large on f_X and small on f_Y. The class $\mathcal{F}$ must be "rich enough" to uniquely identify $f_X = f_Y$ a.e..

Let us consider the space of continuous and bounded functions on Ω, i.e., $\mathcal{F} \equiv C^0(\Omega)$, then [GBR$^+$12, Lemma 1] implies the identity of indiscernibles. Nevertheless, this class of functions is too rich to be practical and the "richness" of the class affects the convergence rate of estimates [GBR$^+$12].

Reproducing kernels are a bridge from linearity to non-linearity for algorithms that can be expressed in terms of dot products, by mapping the original space to a higher (even infinite) dimensional one. Fortunately, this mapping does not need to be explicitly computed as only a *reproducing kernel function* is required to replace the dot product used by the algorithm (the so-called *kernel trick*).

One way to obtain a class that is rich enough to identify any kind of dissimilarity but simple enough to have good convergence rates is to consider a *reproducing kernel Hilbert space* (RKHS) $\mathcal{H}$, which is a *Hilbert space* (i.e., an abstract vector space with inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$) of functions $g : \Omega \to \mathbb{R}$ equipped with the linear evaluation functional $\delta_x : g \to g(x)$ that is bounded on $\mathcal{H}$. Then, the Riesz representation theorem ensures that $\forall x \in \Omega$ there exists a unique $\phi_x \in \mathcal{H}$, called the *feature mapping* such that $g(x) = \langle g, \phi_x\rangle_{\mathcal{H}}$. The reproducing kernel is the (positive definite) function $k : \Omega \times \Omega \to \mathbb{R}$ defined as $k(x,y) = \langle\phi_x, \phi_y\rangle_{\mathcal{H}}$.

Then, the unit ball (i.e., functions g such that $\|g\| = \sqrt{\langle g, g\rangle_{\mathcal{H}}} \leq 1$) in a RKHS is a more restrictive class, but yet rich enough to uniquely identify whether $f_X = f_Y$ [GBR$^+$12, Theorem 5] when a *universal* kernel is used. Furthermore, the MMD satisfies all the metric properties except non-negativity when $\mathcal{H}$ is a universal RKHS defined on a compact metric space, which is the case for the Gaussian

$$k(x,y) \equiv \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \tag{1.8}$$

and Laplace

$$k(x,y) \equiv \exp\left(-\frac{\|x-y\|}{\sigma}\right) \tag{1.9}$$

cases. [GBR$^+$12, Lemma 6] provides an expression for squared MMD in terms of the reproducing kernel functions in a RKHS:

$$\mathrm{MMD}^2[\mathcal{F}, f_X, f_Y] = \mathbb{E}_{X,X'}\left[k(X,X')\right] - 2\mathbb{E}_{X,Y}\left[k(X,Y)\right] + \mathbb{E}_{Y,Y'}\left[k(Y,Y')\right] \tag{1.10}$$

where X' is an independent copy of X and Y' is an independent copy of Y.

Another interesting special case of this measure is obtained for $\Omega = \mathbb{R}$, when considering in Eq. (1.7) the class of functions with bounded variation, i.e., functions g such that $\int|\partial_x g(x)|dx \leq C$ for some constant C. Setting $C = 1$ yields the Kolmogorov-Smirnov metric [GBR$^+$12, Proposition 21] that yields the famous Kolmogorov-Smirnov two-sample test [Kol33, Smi48].

As shown in [GBR$^+$12], the Monge-Wasserstein, or Mallows, distance can also be obtained as a special case of MMD using some specific function class in Eq. (1.7). Informally, if we see the two densities as different ways of piling up a certain amount of dirt over Ω, the Monge-Wasserstein distance is the minimum cost of turning one pile into the other, where the cost is the amount of dirt moved times the distance by which it is moved.

1.2.2 Binary Classification

As we will see throughout this work, statistical dissimilarity is intimately related to *binary classification*. For this task, it is assumed that samples are drawn from a mixture of the two distributions and they are labeled according to the distribution they were drawn from: the label 0 for f_X and the label 1 for f_Y.We denote by L the random variable corresponding to the label and $\theta_0 \equiv \mathbb{P}(L=0)$. The label alphabet is denoted as $\mathcal{A} \equiv \{0,1\}$. Therefore, a pair of random variables (Z,L), corresponding respectively to position (whose components are usually called *features*) and label, with joint density $f_{Z,L}$, is considered. Since Z comes from a mixture of X and Y, it takes values on Ω as well. The density of each population can be expressed as $f_X(z) = f(z|L=0)$ and $f_Y(z) = f(z|L=1)$. The density of Z is $f_Z \equiv \theta_0 f_X + (1-\theta_0) f_Y \equiv \bar{f}$.

Then, having seen a set of samples with their labels and the value of Z of a new sample, the goal is to predict its label L either in *hard* way by telling one value or in a *soft* way by estimating a probability distribution on its value while trying to minimize some *loss function*. Intuitively, the higher the dissimilarity between f_X and f_Y is, the easier the classification becomes.

Hard Classifiers and the 0/1 Loss A *hard classifier* is a function $g : \Omega \to \mathcal{A}$. An *error* occurs if $g(Z) \neq L$. The *0/1 loss* is defined as $\mathcal{L}_{0/1}(g(z), l) \equiv \mathbb{1}_{l \neq g(z)}$, where $\mathbb{1}$ denotes the indicator function. The *Bayes error* is given by the function $g^\star$ that minimizes the probability of error, i.e.

$$\mathcal{L}^\star_{0/1} \equiv \min_{g:\Omega\to\mathcal{A}} \mathbb{P}(g(Z) \neq L). \tag{1.11}$$

The mutual information $I(Z;L)$ quantifies how much the knowledge of Z reduces the uncertainty about L. The JSD between f_X and f_Y turns out to be equal to $I(Z;L)$ since, denoting by $H(Z)$ the entropy of Z and by $H(Z|L)$ the conditional entropy of Z given L, we have that

$$\begin{aligned} I(Z;L) &= H(Z) - H(Z|L) && (1.12)\\ &= H(f_Z) - \theta_0 H(f_X) - (1-\theta_0) H(f_Y) && (1.13)\\ &\equiv JS_{\theta_0}(f_X \| f_Y). && (1.14) \end{aligned}$$

Furthermore, the JSD provides the following bounds on the Bayes error rate [Lin91]:

$$\frac{1}{4}\left(h(\theta_0) - JS_{\theta_0}(f_X\|f_Y)\right)^2 \leq \mathcal{L}^\star_{0/1} \leq \frac{1}{2}\left(h(\theta_0) - JS_{\theta_0}(f_X\|f_Y)\right) \tag{1.15}$$

where $h(\cdot)$ denotes the binary entropy (i.e., $h(\theta) \equiv -\theta\log_2\theta - (1-\theta)\log_2(1-\theta)$).

As noted in [GBR+12, Remark20][Fri04] and further discussed in [RW11], the optimal classifier $g^\star$ (or some sequence of classifiers that approaches it), can be turned into a dissimilarity measure by considering $\mathbb{E}_X[g^\star(X)] - \mathbb{E}_Y[g^\star(Y)]$.

Soft Classifiers and the Logarithmic Loss A *soft classifier* (also known as *probabilistic classifier*) is a function $g : \Omega \to \mathcal{D}_\mathcal{A}$ where $\mathcal{D}_\mathcal{A}$ is the set of all distributions over $\mathcal{A}$ (i.e., $\mathcal{D}_\mathcal{A} \equiv \{p : \sum_{l\in\mathcal{A}} p(l) = 1\}$). Of particular interest for our work is the *logarithmic (log) loss*, which is defined as $\mathcal{L}_{\log}(g(z), l) \equiv -\log g(z)(l)$.

A vast literature in Information Theory exists for this loss (see, e.g., [CT06]). The log loss possesses the following important properties (see, e.g., [MF98] for more details):

- it is monotonically decreasing with the assigned probability to the observed label,
- technically, it is convenient to work with since it converts joint probabilities, or equivalently, products of conditional probabilities into cumulative sums, and
- it can be used for the purpose of testing the validity of statistical models (i.e., families of parametric distributions) by Dawid's prequential principle [Daw84] encompassed in Rissanen's Minimum Description Length principle [Ris78, Ris96, Ris87, Grü07]. One of the reasons for this is that the loss is minimized, in the expected or in the almost sure sense, by the distribution that generates the samples, a property shared by very specific loss functions (see [MF98, MGS93]).

This last property is particularly important since it means that when learning a soft classifier one usually seeks to approximate the true conditional distribution $g^\star(z)(\cdot) \equiv \mathbb{P}(L = \cdot | Z = z)$. This makes the problem addressable by *nonparametric regression with random design* (see, e.g., [GK02]), where given a random vector (Z, R), where $Z \in \mathbb{R}^d$ and the response variable $R \in \mathbb{R}$, the goal is to estimate the *regression function* defined as

$$m(z) \equiv \mathbb{E}[R|Z = z]. \tag{1.16}$$

Therefore, setting $R \equiv L$ yields

$$m(z) = \mathbb{P}(L = 1|Z = z) \equiv g^\star(z)(1). \tag{1.17}$$

1.3 Learning Dissimilarity Measures and Classifiers

In the previous section, we have introduced different measures of dissimilarity and the related problem of classification. In this section, we will address the problem of learning the involved objects, in particular, dissimilarity measures and classifiers (with emphasis on soft ones).

1.3.1 Settings

We first present some of the common settings for learning dissimilarity measures and classifiers.

Dissimilarity Measures

When learning dissimilarity measures, two i.i.d. samples are usually given, that is $x_1, \ldots, x_{n_0}$ and $y_1, \ldots, y_{n_1}$ whose corresponding random variables $X_i \in \Omega$ and $Y_i \in \Omega$ are i.i.d. with densities f_X and f_Y respectively.

Classification

For the classification problem, different settings can be considered. When the construction uses label information, it is called *supervised learning*. In this problem, samples (Z_i, L_i) are usually assumed to be i.i.d. with joint density $f_{Z,L}$.

Batch The *batch* (or *off-line*) learning setting is the classical case of supervised learning (see, e.g., [DGL96]). A *training set* of n labeled samples is given, that is $(z, l)^n \equiv (z_1, l_1), \ldots, (z_n, l_n)$ from which a *classifier* $\hat{g}_n : \Omega \to \mathcal{A}$ is built. Then, this classifier is assessed on a different *test set* $(z_{n+1}, l_{n+1}), \ldots, (z_{n+m}, l_{n+m})$. Note that we will also refer to the pair sequence $(z, l)^n$ as l^n, z^n.

On-line In the *on-line* setting, symbols are observed sequentially one by one. Based on previously observed pairs $(z_1, l_1), \ldots, (z_{i-1}, l_{i-1})$, an on-line learning algorithm tries to build a *classifier* $\hat{g}_{i-1} : \Omega \to \mathcal{A}$ predict the next label L_i to come given its position z_i.

Therefore, an on-line learning algorithm can improve its performance as new samples are seen. This is also known as *prediction with side information* (see, e.g., [CB06]). Notice that this is also a case of supervised learning.

Semi-supervised In the *semi-supervised* setting, the classifier has access to an additional set of unlabeled samples $(z'_1, \ldots, z'_u)$. This can be of interest since it is common in practice that only few labeled samples are available for training while a larger quantity of unlabeled samples are available.

Roughly speaking, the semi-supervised learning is useful, when the unlabeled data helps elucidating the structure of the classifier. In other words, the knowledge on f_Z acquired through the unlabeled data has to carry information that is useful in the inference of $\mathbb{P}(L = \cdot | Z = z)$. This can be the case under certain assumptions (see, e.g., [CSZ+06] for an in-depth treatment of the subject). For example, if we know that the points of each class tend to form clusters, then the unlabeled data can help in finding the boundary of each cluster more accurately. A simple method would be to identify clusters with some *unsupervised* method (without taking into account any label) and then assigning a label to each cluster according to the given labels.

1.3.2 Desired properties

Consistency When estimating dissimilarity measures, one usually seeks some sort of consistency, that is, that estimated measure of dissimilarity converges (in some sense) to the true dissimilarity measure as the number of samples used for the estimation tends to infinity.

For a soft classifier $\hat{g}_n(z)$ to be consistent, it is usually required that it converges is some sense to $\mathbb{P}(L = \cdot|Z = z)$. For example, using a *nonparametric regressor* $m_n(z)$ (i.e., an estimate of $m(z)$ built using a set of n samples) the following strong sense known as *strong pointwise consistency* can be considered (see, e.g., [GK02])

$$m_n(z) \xrightarrow{n\to\infty} m(z) \text{ a.s.} \tag{1.18}$$

for f_Z-almost all z. Using the L_2 norm to quantify the difference or *error* between the estimate and the true regression function, two other notions of consistency are commonly used, which are the *strong consistency*

$$\mathbb{E}_Z\left[|m_n(Z) - m(Z)|^2\right] \xrightarrow{n\to\infty} 0 \text{ a.s.} \tag{1.19}$$

and the *weak consistency*

$$\mathbb{E}_{L^n,Z^n}\left[\mathbb{E}_Z\left[|m_n(Z) - m(Z)|^2\right]\right] \xrightarrow{n\to\infty} 0. \tag{1.20}$$

These definitions translate straightforwardly to conditional probability estimation using the mapping of Eq. (1.17).

Bias-Variance Tradeoff The bias-variance tradeoff is the dilemma that is faced when trying to simultaneously minimize two sources of error that prevent supervised learning algorithms from generalizing beyond their training set:

- the *bias* is the error generated by erroneous assumptions in the learning algorithm. High bias can cause *underfitting*, i.e., missing relevant relations between features Z and labels L.
- the *variance* is the error generated by the sensitivity of the learning algorithm to small fluctuations. High variance can cause *overfitting*, i.e., capturing noise or relations that do not generalize.

For example, in nonparametric regression, let m_n be an arbitrary estimate, then for any $Z \in \mathbb{R}^d$ the expected L_2 error of m_n at Z can be written as [GK02]

$$\begin{aligned}\mathbb{E}_{L^n,Z^n}\left[\mathbb{E}_Z\left[|m_n(Z) - m(Z)|^2\right]\right] &= \mathbb{E}_Z\left[\mathbb{E}_{L^n,Z^n}\left[|m_n(Z) - m(Z)|^2\right]\right] &(1.21)\\ &= \mathbb{E}_Z\left[\mathrm{Var}(m_n(Z))\right] + \mathbb{E}_Z\left[|\mathrm{Bias}(m_n(Z))|^2\right] &(1.22)\end{aligned}$$

where $\mathrm{Var}(m_n(z))$ is the variance of the random variable $m_n(z)$ and $\mathrm{Bias}(m_n(z))$ is the difference between the expectation of $m_n(z)$ and $m(z)$, in both cases with respect to the random variables L^n, Z^n.

As shown in [GK02, Sec. 2.3], the importance of these decompositions is that the expected variance and the expected squared bias depend in opposite ways on the "wiggliness" ("non-smoothness") of an estimate, yielding a bias-variance tradeoff. In other words, wiggly estimates allow to better fit the data and thus to reduce expected bias but, on the other hand, it is compensated by an increase in expected variance.

Universality Let us define, in the online case, the *redundancy* of a soft classifier $\hat{g}_n(z)$ as the accumulated *excess loss* with respect to the true conditional distribution, i.e.,

$$R_n\left(\hat{g}, g^\star\right) \equiv \sum_{i=1}^{n} \mathcal{L}_{\log}(\hat{g}_{i-1}(z_i), l_i) - \mathcal{L}_{\log}(g^\star(z_i), l_i). \tag{1.23}$$

Then, a desired property is that the normalized redundancy $\frac{1}{n}R_n\left(\hat{g}, g^\star\right)$ vanishes to zero in some sense (e.g., in expected value or almost surely), as n tends to infinity [Grü07, Def. 2-3, p. 200]. Naturally, the convergence speed, which reflects the learning rate, is important too.

Note that in the literature, the term "universal" is also used in the non-stochastic case, where prediction is also done using distributions but the goal is to approach the performance of the best distribution of some given reference class (see, e.g., [CB06]).

Effectiveness in High-Dimensions The traditional bane of nonparametric statistics is known as the *curse of dimensionality* [Bel61], meaning that the prediction performance deteriorates dramatically as the dimension d of the data increases. In nonparametric regression, where $m(x)$ is p times continuously differentiable, the best convergence rate of the expected error in norm L_2 for the worst case of $m(x)$ (i.e., the *minimax rate*) is $n^{\frac{-2p}{(2p+d)}}$ (see [Sto80, Sto82]).

In other words, the number of samples required to attain a low expected error may be exponential in d, which can be difficult to satisfy even for moderate values of d. However, it is often the case that data which appears high dimensional has, in fact, some lower *intrinsic dimensionality* (in some sense), due to the fact that, roughly speaking, features may be correlated or redundant. This is the case in many real life data, e.g., in human body motion capture where 100 markers can be attached to the body and each marker measures position in three dimensions: dimension is 300 but motion is in fact constrained by joints and angles which yields data of much lower intrinsic dimension (see, e.g., [DF08, VKD09]).

Therefore, methods whose error convergence rate depend only on the intrinsic dimension are of special interest.

Algorithmic Complexity Online methods ideally need to have constant processing time and memory in order to be effective in practice for potentially unbounded sample sizes. Nevertheless, this can be difficult to achieve when consistency and universality are sought.

1.3.3 Methods of Estimation

Now we present three classic and widely used classes of methods for building estimates for dissimilarity measures and for conditional probability, namely partitioning, nearest-neighbors and kernel based methods.

For conditional probability estimation via nonparametric regression, *local averaging estimators* of the following form can be used to estimate the regression function $m(z)$ (Eq. (1.16)) (see, e.g., [GK02]):

$$m_n(z) \equiv \sum_{i=1}^{n} W_{n,i}(z) R_i \tag{1.24}$$

where the weights $W_{n,i}(z) \equiv W_{n,i}(z, Z_1, \ldots, Z_n) \in \mathbb{R}$ depend on $Z_1, \ldots, Z_n$.

In the case of the estimation of the $\mathfrak{f}$-divergence , a traditional approach is to first estimate the individual densities using some nonparametric method (e.g., kernel density estimation) and substitute in the definition formula. Nevertheless, the problem of estimating the individual densities is harder than estimating the divergence (see, e.g., [PC08]), and thus, by Vapnik's advice [Vap99], should be avoided.

Partitioning

This class of methods is based on considering a sequence of (finite or countably finite) partitions $\mathcal{P}_n = \{A_{n,1}, A_{n,2}, \ldots\}$ of $\mathbb{R}^d$, where $A_{n,1}, A_{n,2}, \ldots$ are Borel sets.

Classification In order to build a locally averaging estimator, consider the following functions

$$W_{n,i}(z) \equiv \frac{\mathbb{1}_{Z_i \in A_{n,j}}}{\sum_{l=1}^{n} \mathbb{1}_{Z_l \in A_{n,j}}} \text{ for } z \in A_{n,j}. \tag{1.25}$$

The sequence of partitions is *nested* if the sequence of generated σ-algebras $\mathcal{F}(\mathcal{P}_n)$ is increasing (i.e., it is a *filtration*).

For $z \in \mathbb{R}^d$, let us define $A_n(z) \equiv A_{n,j}$ if $z \in A_{n,j}$, .i.e., the cell of the n-th partition containing z. Denoting by λ the Lebesgue measure on $\mathbb{R}^d$ and defining the *diameter* of a cell as

$$\operatorname{diam}(A_n(z)) \equiv \sup_{u,v \in A_n(z)} \|u - v\| \tag{1.26}$$

the following result proves the convergence (see [GK02, Thm. 24.3 and Thm. 25.6])

Theorem. 1.1. *Consider the estimator obtained from Eq. (1.24) and 1.25. If the sequence of partitions $\mathcal{P}_n$ is nested, $diam(A_n(z)) \xrightarrow{n\to\infty} 0$ for each $z \in \mathbb{R}^d$ and*

$$\sum_{i=n}^{\infty} \lambda(A_i(z)) = \infty \tag{1.27}$$

for each z and n, then m_n is weakly and strongly consistent. Moreover, if, for all z,

$$\frac{n\lambda(A_n(z))}{\log n} \to \infty \tag{1.28}$$

and $|R| < C$ for some $C < \infty$ then m_n is also strongly pointwise consistent.

The difficult part of using partition based methods is to define how the space should be partitioned in order to satisfy conditions that makes the procedure consistent but also to be suitable for high-dimensional data with low-intrinsic dimension.

Regression trees proposed in [GO80, GO84] use a data dependent rule to split the space recursively yielding therefore a structure that can be represented as a tree , thus enjoying logarithmic time to find a cell when the tree is close to balanced.

Many extensions and variations to these idea have been proposed including the popular Breiman's Random Forests [Bre01], online versions as in [DMFN13] and the more recent intrinsic-dimension adaptive Random Projection Trees [DF08, KD12]. In contrast to this intrinsic-dimension adaptive partitioning, axis-parallel splitting rules have been shown to be prone to the curse of dimensionality even when intrinsic dimension is lower [DF08].

In [Nob96], conditions that guarantee strong consistency of regressors based on data-dependent partitions are shown.

Divergence In [WKV05], the KL divergence is estimated by estimating the Radon-Nikodym derivative using the ratio of the empirical measures $P_{X,n_0}(A_{n,j})$ and $P_{Y,n_0}(A_{n,j})$ in each cell $A_{n,j}$ of each partition of the sequence, i.e.,

$$P_{X,n_0}(A_{n,j}) \equiv \frac{1}{n_0} \sum_{i=1}^{n_0} \mathbb{1}_{X_i \in A_{n,j}} \tag{1.29}$$

$$P_{Y,n_1}(A_{n,j}) \equiv \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbb{1}_{Y_i \in A_{n,j}}. \tag{1.30}$$

The partitions are built using *statistically equivalent blocks* (Gessaman's partition [Ges70, Nob96]), with respect to the reference measure P_Y, using the following procedure. First, the intended number of points per cell k_n for the n-th partition must be chosen. Then, $C_n \equiv \lfloor (n/k_n)^{1/d} \rfloor$ is the number of cells for the n-th partition. The iterative part starts with one cell containing all the points and then, for every coordinate $i = 1..d$, the following step is repeated:

- For each cell, project the samples along the i-th axis and split it into C_n cells, using hyperplanes perpendicular to the i-th axis, such that all of them contain the same number of points (with the possible exception of the last cell)

The almost sure convergence of this method was proved under mild conditions. The authors propose some extensions that improve convergence rate by using non-uniform data-adaptive partitioning schemes and using a bias correction term.

This method was further analyzed (for multivariate data) and extended in [SN07, SN10] for more general partitioning schemes. More precisely, some asymptotic conditions on the maximum number of points per cell, the complexity of the partition rule (related to Vapnik-Chervonenkis complexity [Vap99]) and the cells' diameters were shown to guarantee the almost sure consistency of the estimator.

Nearest Neighbors

This class of methods is based on considering a number of neighbor points to estimate the quantity of interest at each point $z \in \mathbb{R}^d$.

Classification In order to build a locally averaging estimator, consider the following weighting functions

$$W_{n,i}(z) = \begin{cases} 1/k_n & \text{if } Z_i \text{ is among the } k_n \text{ nearest neighbors of } z \\ 0 & \text{otherwise} \end{cases} \tag{1.31}$$

where k_n is a parameter that can depend on n and the notion of nearest depends on the chosen distance and some tie-breaking rule.

The following theorem gives conditions to guarantee strong pointwise consistency [GK02, Thm. 25.17]:

Theorem. 1.2. *If $|R| < C$ for some $C < \infty$,*

$$\frac{k_n}{\log n} \to \infty \text{ and } \frac{k_n}{n} \to 0,$$

then the k_n-NN estimate using Euclidean distance is strongly pointwise consistent.

In [Kpo11], it was shown that for regression functions that are λ-Lipschitz, the rate of convergence of a k_n-NN regressor depends only on the intrinsic dimension. Moreover, the authors propose a method to locally choose k, which guarantees nearly optimal minimax rate.

The drawback of the exact nearest neighbors approach is that the number of neighbors need to grow with n, thus making it inappropriate in an online setting.

Divergence In [WKV09, PC08], the KL divergence is estimated by using a k-nearest-neighbor approach similar to the one use for density estimation. For each sample x_i, consider the volume V_{x_i} of the ball B centered at x_i enclosing the k-nearest neighbors, then the density estimate is:

$$\hat{f}_0(x_i) \equiv \frac{k}{n_0 V_{x_i}}. \tag{1.32}$$

For consistent density estimation a growing k is required. With respect to the Bias-Variance tradeoff for finite sample sizes, as discussed in [Dev87, Sil86], a smaller k leads to a lower bias and a higher variance.

In [PC08], it is shown that a a fixed k guarantees almost sure consistency of the KL divergence estimation while empirically outperforming methods based on consistent density estimation. In [WKV09], it is shown that using an adaptive k (as a result of fixing a radius instead of fixing k) reduces the bias in the estimation.

Kernel

As a dot product, a kernel function can be seen as a similarity measure between points (see, e.g., [SS02]). Generalizing this idea, a *kernel function* is a weighting function that can be used to define a weighted neighborhood of a given point.

Classification The *Nadaraya-Watson kernel estimate* (see, e.g., [GK02]) is a locally averaging estimator that uses the following weighting functions

$$W_{n,i}(z) \equiv \frac{k_n(z, Z_i)}{\sum_{j=1}^{n} k_n(z, Z_j)} \tag{1.33}$$

where the kernel functions k_n have the form $k_n(z, Z_i) \equiv K\left(\frac{z - Z_i}{h_n}\right)$ where the *bandwidth* h_n depends only on the sample size. Usually, $K(x)$ is symmetric with respect to 0, nonnegative and is large when $\|x\|$ is small. For example, the *naive* (or *window*) kernel is defined as follows

$$K(x) \equiv \mathbb{1}_{\|x\| \leq 1}. \tag{1.34}$$

Stone's theorem [Sto77] allows to prove weak convergence under general conditions on h_n and K (see [GK02, Sec. 5.2]). The following theorem gives conditions that guarantee strong pointwise consistency when using the naive kernel [GK02, Thm. 25.11].

Theorem. 1.3. *For the naive kernel, assume that* $h_n \xrightarrow{n\to\infty} 0$ *and* $nh_n^d/\log n \xrightarrow{n\to\infty} 0$ *and* $|R| < C$ *for some* $C < \infty$*, then the kernel estimate is strongly pointwise consistent.*

An important technique that is worth mentioning is the popular *Support Vector Machine (SVM)* which is meant for hard classification (see [SS02]). Using the kernel trick, the method looks for the hyperplane that best separates the two populations in the higher-dimensional space which allows non-linear separation in the original space. The distance of a new point to be classified to this hyperplane can be interpreted as a measure of confidence of the classification. After calibrating according to the *margin* (i.e., the separation of the two populations yielded by the hyperplane), these scores can be converted to values in $[0, 1]$ using Platt Scaling [P+99], which learns a logistic regression model on the scores. As discussed in [LZ05], no guarantee on the quality of this estimate as a probability can be guaranteed, essentially because the margin is not a sufficient statistic.

Alternatively, in [LZ05], a probing method to convert any hard classifier into a soft one with guarantees (i.e., a small error for hard classification implying an accurate estimation of the label probabilities) has been proposed.

In [KSW04], online kernels methods applied, in particular, to classification and regression have been proposed.

Divergence An important kernel related divergence measure that yields a powerful two-sample test is the MMD that we introduced in Section 1.2.1.

A biased estimate is obtained from Eq. (1.10) by substituting expected values with samples averages [GBR+12, Lemma 6]

$$\mathrm{MMD}_b[\mathcal{F}, X, Y] \equiv \left[\frac{1}{n_0^2} \sum_{i,j=1}^{n_0} k(x_i, x_j) - \frac{2}{n_0 n_1} \sum_{i,j=1}^{n_0,n_1} k(x_i, y_j) + \frac{1}{n_1^2} \sum_{i,j=1}^{n_1} k(y_i, y_j) \right]^{\frac{1}{2}} \tag{1.35}$$

which cost $O\left((n_0 + n_1)^2\right)$ to compute. This biased estimate convergences in probability at rate $O\left((n_0 + n_1)^{-\frac{1}{2}}\right)$ to its population value.

Alternative quadratic and linear unbiased estimators for MMD^2 are also presented in [GBR+12]. Although experimentally of slower convergence, the linear alternative is of particular interest for streaming computations since it only requires $O(1)$ memory, while the quadratic versions require $O(n_0 + n_1)$ to compute all interactions.

[RW11] elucidates interesting connections between MMD and SVM, where they also show that the further apart the distributions are, the easier the discrimination problem becomes.

1.3.4 Universal Prediction Methods for Soft Classification

One way of building a universal soft classifier is by considering a sequence of nested *parametric models*, i.e., sets of parametric distributions on infinite sequences $s^\infty \equiv s_1, s_2, \ldots$ where $s_i \in \mathcal{S} \subseteq \mathbb{R}^d$, such that $\mathcal{M}_1 \subseteq \mathcal{M}_2 \subseteq \ldots$ where $\mathcal{M}_i = \{p_\theta : \theta \in \Theta_i \subseteq \mathbb{R}^d\}$ (see, e.g., [Grü07]). Let us define $\mathcal{M} \equiv \cup_{i=1}^\infty \mathcal{M}_i$ If we denote by $\mathcal{M}^\star$ the model to which the true distribution $p^\star \in \mathcal{M}^\star$ belongs, there are two cases of special interest :

- the parametric case: $\mathcal{M}^\star \subseteq \mathcal{M}_k$ for some k,
- the nonparametric but approximable case: if we define the *information closure* $\langle \mathcal{M} \rangle$ as the set of distributions for infinite sequences on $\mathcal{S}$ that can be arbitrarily well approximated by elements of $\mathcal{M}$, i.e., $\langle \mathcal{M} \rangle \equiv \{p^\star | \inf_{p \in \mathcal{M}} D(p^\star \| p) = 0\}$, we have that $\mathcal{M}^\star \setminus \mathcal{M} \neq \emptyset$ and $\mathcal{M}^\star \subset \langle \mathcal{M} \rangle$.

In our case, one could set $\mathcal{X} \equiv \Omega \times \mathcal{A}$ and use joint distributions. Nevertheless, as noted in [vEGdR12, p. 382], for the methods we consider next, it is equivalent to set $\mathcal{X} \equiv \mathcal{A}$ and use conditional distributions instead.

Next, we present two methods to achieve universality.

Bayesian Distributions

In the Bayesian approach, a *prior distribution* represents one's "belief" on which of the distributions of $\mathcal{M}$ generates the data. Then, as data are seen, this prior is updated to obtain a posterior distribution. This posterior can then be used to predict future data (see, e.g., [Grü07, p. 73]).

In the case of a sequence of nested models, a prior $w(\cdot)$ on the models $\mathcal{M}_i$ of the sequence and a prior $\pi(\cdot)$ on the parameter space Ω_i of each model are considered in order to mix the models in the following way:

$$P_{\text{Bayes}(\mathcal{M})}(s^n) \equiv \sum_i w(i) P_{\text{Bayes}(\Omega_i)}(s^n) \tag{1.36}$$

with

$$P_{\text{Bayes}(\Theta_i)}(s^n) \equiv \int_{\theta \in \Theta_i} \pi_i(\theta) p_\theta(s^n) d\theta. \tag{1.37}$$

Using Bayes' formula, one obtains the following predictive formulas:

$$P_{\text{Bayes}(\mathcal{M})}(s_{n+1}|s^n) = \sum_i w(i|s^n) P_{\text{Bayes}(\Theta_i)}(s_{n+1}|s^n) \tag{1.38}$$

with

$$w(i|s^n) = \frac{w(i) P_{\text{Bayes}(\Theta_i)}(s^n)}{\sum_i w(i) P_{\text{Bayes}(\Theta_i)}(s^n)} \tag{1.39}$$

and

$$P_{\text{Bayes}(\Theta_i)}(s_{n+1}|s^n) = \int_{\theta \in \Theta_i} \pi_i(\theta|s^n) p_\theta(s_{n+1}|s^n) d\theta \tag{1.40}$$

with

$$\pi_i(\theta|s^n) = \frac{\pi_i(\theta) p_\theta(s^n)}{\int_{\theta \in \Theta_i} \pi_i(\theta) p_\theta(s^n) d\theta}. \tag{1.41}$$

In order to show that $P_{\text{Bayes}(\mathcal{M})}$ is universal in the parametric case, we denote the true distribution as $p_{\theta^\star} \in \mathcal{M}_k$ for some k and observe that

$$P_{\text{Bayes}(\mathcal{M})}(s^n) \equiv \sum_i w(i) P_{\text{Bayes}(\Theta_i)}(s^n) \geq w(k) P_{\text{Bayes}(\Theta_k)}(s^n) \tag{1.42}$$

which implies that

$$-\log P_{\text{Bayes}(\mathcal{M})}(s^n) \leq -\log w(k) - \log P_{\text{Bayes}(\Theta_k)}(s^n). \tag{1.43}$$

Then, in [LY00], it was shown that for $p_{\theta^\star}$ belonging to an exponential family with k-parameters (here, we assume, for simplicity, that the index in $\mathcal{M}_k$ correspond to the number of parameters), with $p_{\theta^\star}$-probability 1,

$$R_n\left(P_{\text{Bayes}(\Theta_k)}, p_{\theta^\star}\right) \equiv -\log P_{\text{Bayes}(\Theta_k)}(s^n) + \log p_{\theta^\star}(s^n) \tag{1.44}$$

$$\leq \frac{k}{2} \log \frac{n}{2\pi} - \log \pi_k(\theta^\star) + \log \sqrt{\det I(\theta^\star)} + C_n \log\log n + o(1) \tag{1.45}$$

where $I(\cdot)$ denotes the Fisher information matrix and C_n is a random variable such that with probability 1 is nonnegative for all n, is upper-bounded by k for all n, and that takes on a value ≥ 1 for infinitely many n. Combining Eqs. (1.43) and (1.44) yields that $\frac{1}{n} R_n\left(P_{\text{Bayes}(\mathcal{M})}, p_{\theta^\star}\right)$ is upper bounded by a quantity that vanishes to zero almost surely. It is important to note that the convergence rate depends on the number k of parameters of the true distribution.

In the non-parametric case, the universality of $P_{\text{Bayes}(\mathcal{M})}$ has been proved for some model classes under certain conditions (see [Grü07, Section 13.4.1]).

The Switch-Distribution

Universality of Bayesian distributions stems from the fact that their performance is asymptotically as good as any of the distributions under consideration. Nevertheless, they suffer from the *catch-up phenomenon*, i.e., their convergence rate is not optimal. In [vEGdR12], it has been shown that a better convergence rate can be achieved by allowing models to change over time, i.e., by considering instead of a set of distributions $\mathcal{M}$ a (larger) set constituted by sequences of distributions of $\mathcal{M}$. This can be explained intuitively by the fact that when a small number of samples has been seen, models with fewer parameters work better than more complex ones even when the true model is more complex. Therefore, it can be beneficial to consider distributions that *switch* from simpler distributions to more complex ones. The distribution has a still a Bayesian form but the mixture is done over sequences of models.

1.4 Detection: Hypothesis Testing

In this section, we recall fundamental concepts used in statistical hypothesis testing with a focus on nonparametric two-sample testing.

1.4.1 Hypothesis Testing

Statistical tests allow to make decisions using samples by means of analyzing some global property—a *statistic*—, which is a function of them, and estimating the probability that the observed property value has occurred by chance alone.

Hypotheses In classical hypothesis testing, the setting is asymmetric. The null hypothesis $\mathtt{H}_0$ is meant to represent the belief that the statistician is trying to refute. The probability of observing the global property value by chance alone is therefore computed assuming the null hypothesis is true. The alternative hypothesis $\mathtt{H}_1$ represents the class of situations a test would be sensitive to in order to refute the null hypothesis.

When the hypotheses have a parametric form (with a parameter θ), i.e.

$$\begin{cases} \mathtt{H}_0 & : \theta \in \Theta_0 \\ \mathtt{H}_1 & : \theta \in \Theta_1 \end{cases}, \tag{1.46}$$

two types of hypotheses can be considered. Hypotheses that are completely specified, i.e., $|\Theta_\cdot| = 1$, are *simple*, otherwise they are *composite*.

Nonparametric hypotheses include unknown or infinite parameters and therefore are an extreme case of composite hypotheses.

Statistic A *statistic* T_n is a random variable that is a function of the observed samples, aiming at revealing the evidence against the null hypothesis.

In order to provide evidence against the null hypothesis, the statistic must take extreme values (either large or small).

p-value A *valid p-value* $p \in [0, 1]$ is a statistic such that

$$\mathbb{P}_{\mathtt{H}_0}(p \leq \alpha) \leq \alpha, \ \forall \alpha \in [0, 1]. \tag{1.47}$$

An *exact p-value* is uniform under the null hypothesis, i.e.,

$$\mathbb{P}_{\mathtt{H}_0}(p \leq \alpha) = \alpha, \ \forall \alpha \in [0, 1]. \tag{1.48}$$

An exact p-value can be interpreted as the probability of observing a statistic at least as extreme as the one actually observed, assuming that the null hypothesis is true, where "more extreme" depends on the way the hypothesis is tested.

Rejection under significance level α The null hypothesis is rejected with significance α if there exists a threshold T_α such that $\mathbb{P}_{\mathtt{H}_0}(T_n \leq T_\alpha) \leq \alpha$ (we assumed that a small value of the statistic represents stronger evidence for rejecting the null hypothesis).

When using a p-value, this is equivalent to rejecting when p is under the threshold α, since $\mathbb{P}_{\mathtt{H}_0}(p \leq \alpha) \leq \alpha$. The range of values of T_n for which the null is rejected is called the *rejection region*.

Error types Since these tests are based on a finite number of samples, it is possible that an incorrect answer is returned. A *Type I error* occurs if $\mathtt{H}_0$ is rejected when it is actually true (i.e., a *false positive*). Conversely, a *Type II error* occurs if $\mathtt{H}_0$ is not rejected when it is actually false (i.e., a *false negative*). In general, there is a trade-off between the two error probabilities.

The significance level α is therefore an upper bound on the type I error probability. Nevertheless, when the alternative hypothesis represents a large class of probability distributions, it can be difficult or impossible to provide guarantees for type II error. For this reason, when not enough evidence is found to reject the null hypothesis, we avoid saying that the null hypothesis is accepted.

Power and consistency The *power* of a test is defined as the test's probability of correctly rejecting the null hypothesis (thus, $1 - \mathbb{P}(\text{Type II error})$) which, in other words, is the test's ability of identifying the alternative hypothesis. A test is *consistent* if its power tends to one when the number of samples tends to infinity. When using a p-value, consistency can be expressed as

$$\mathbb{P}_{\mathtt{H}_1}(p \leq \alpha) \xrightarrow{n \to \infty} 1. \tag{1.49}$$

The Neyman-Pearson Fundamental Lemma Ideally, a *Uniformly Most Powerful (UMP)* test is desired, i.e., a test that, for all the distributions considered by the alternative, has the highest power for a given level α.

The following fundamental theorem describes which tests are UMP when both the null and the alternative hypotheses are simple (see, e.g., [CB01, p.388]).

Theorem. 1.4 (Neyman-Pearson Lemma [NP33])**.** *Consider testing* $\mathtt{H}_0 : \theta = \theta_0$ *versus* $\mathtt{H}_1 : \theta = \theta_1$ *given the samples* $x^n = x_1, \ldots, x_n$*, where the densities are* $f_{\theta_0}(\cdot)$ *and* $f_{\theta_1}(\cdot)$ *respectively. Then, the* likelihood ratio test *that rejects* $\mathtt{H}_0$ *when*

$$T_n \equiv \lambda_n \equiv \frac{f_{\theta_0}(x^n)}{f_{\theta_1}(x^n)} \leq \eta \tag{1.50}$$

where

$$\mathbb{P}_{\mathtt{H}_0}(\lambda_n \leq \eta) = \alpha \tag{1.51}$$

is the most powerful test at level α *for a threshold* $\eta \geq 0$*.*

When the alternative hypothesis is composite, it is often not possible to construct UMP tests (see, e.g., [CB01, 8.3.19]).

Sequential Testing In *sequential hypothesis testing* (also known as *sequential analysis*), the sample size is not fixed in advance and samples are processed sequentially until conclusive results are observed in accordance with a pre-defined stopping rule. This allows to potentially reduce the number of samples that need to be collected in order to reach a conclusion, thus reducing costs of data collection (financial, human, etc).

One classical example of such procedure is Wald's Sequential Probability Ratio Test (SPRT) [Wal45], which considers the following simple hypotheses:

$$\begin{cases} \mathtt{H}_0 : & X \text{ has density } f_{\theta_0}(\cdot) \\ \mathtt{H}_1 : & X \text{ has density } f_{\theta_1}(\cdot) \end{cases}. \tag{1.52}$$

Then, given a sequence of i.i.d. observations $x_1, \ldots, x_i, \ldots$, the logarithm of the likelihood ratio is incrementally computed, i.e.,

$$S_i \equiv S_{i-1} + \log \frac{f_{\theta_1}(x_i)}{f_{\theta_0}(x_i)} \tag{1.53}$$

with $S_0 \equiv 0$.

The stopping rule is based on two parameters $-\infty < a < 0 < b < \infty$ that control the desired Type I and Type II error probabilities and is defined as

$$\begin{cases} a < S_i < b & : \text{ continue} \\ S_i \geq b & : \text{ accept } \mathtt{H}_1 \\ S_i \leq a & : \text{ accept } \mathtt{H}_0 \end{cases}. \tag{1.54}$$

In practice, Wald proposes to use $a \equiv \log \frac{\beta}{1-\alpha}$ and $b \equiv \log \frac{1-\beta}{\alpha}$, which guarantee an actual Type I error probability $\alpha' \leq \alpha$ and an actual Type II error $\beta' \leq \beta$. Wald proved that this test finishes with probability one.

Wald's scheme relies on the perfect knowledge of the probability density function for each hypothesis. Recently, in[SMFLPCAR08], this restriction has been somewhat relaxed for Bernoulli distributions, by allowing some uncertainty in the knowledge of the parameter p defining them.

For more material on statistical testing see, e.g., [CB01, LR05].

1.4.2 The Two-sample Problem

Given samples from two distributions, a two-sample test aims at comparing two distributions by statistically asserting whether they are different in some aspect or not. For example, one could ask whether two distributions have the same variance or not, given a set of samples from each. It is also possible to consider more than two samples in which case it is referred as the k-sample problem.

The *two-sample problem* consists in determining whether the two distributions are equal or not. Parametric tests for the two-sample problem use some assumption on the nature of the distributions that generated samples (e.g., Gaussian), while nonparametric ones are meant to detect differences without assumptions. In the most general form, the null and the alternative hypotheses for the two-sample problem are

$$\begin{cases} \mathtt{H}_0 & : f_X = f_Y \text{ a.e.} \\ \mathtt{H}_1 & : \neg \mathtt{H}_0 \end{cases}. \tag{1.55}$$

Desired Properties and Negative Result

Type I Error Control In many cases, authors prove that the obtained tests are only asymptotically of level α. Nevertheless, for small sample sizes, an important feature of nonparametric two-sample tests is whether the distribution of the test statistic is distribution-free (i.e., independent of the sample's distribution) and whether an exact formula is known, in which case it is said that test is exactly of level α.

Another option is to obtain a bound for the threshold (which in terms of p-value would yield a valid p-value) which results in a *conservative* test with a Type I probability error actually lower than the nominal value α and consequently a lower power.

Type II Error and Consistency It is usually desired to have tests with proven consistency. In the nonparametric case, power and convergence speed depend on the nature of the dissimilarity, which makes no test superior than any other in every case.

In fact, even if a test for the two-sample problem is consistent, at a given fixed sample size, it is not possible to provide guarantees on the Type II error, without prior assumptions on the nature of the difference between the two distributions (see [GBR+12] for a proving example).

Complexity Another important aspect, especially for high-dimensions and when the sample size is large, is the computational complexity involved in the calculation of the test statistic and its threshold.

Classical Nonparametric One-dimensional Two-sample Tests

The classical univariate Wald-Wolfowitz runs test is based on ordering the samples of the two populations and counting the number of runs of samples of the same population. A more powerful classical test is the one of Kolmogorov-Smirnov which is based on estimating the largest difference between the two unknown cumulative distributions. More precisely, the test statistic is $D_{n_0,n_1} := \max_x |\hat{F}_X(x) - \hat{F}_Y(x)|$, where $\hat{F}_X$ and $\hat{F}_Y$ are the empirical cumulative distributions.

The classical Mann-Whitney U test [MW47] aims at detecting whether one random variable X tends to have larger values than another variable Y. More precisely, the test is consistent against the following alternative hypothesis: the probability of X exceeding an observation from Y (accounting for ties by adding $\frac{1}{2}P(X = Y)$) is greater than $\frac{1}{2}$ [Leh51], i.e., $P(X > Y) + \frac{1}{2}P(X = Y) > \frac{1}{2}$.

This general alternative implies (see [FP09] for a discussion) more restrictive alternatives like X being stochastically larger than Y (i.e., the cumulative distribution functions satisfy $F_Y(a) > F_X(a)$ for every a) and locations shifts (i.e., $F_Y(a) = F_X(a + \delta), \delta \neq 0$).

The test can be *one-sided*, in which case the alternative is one of the previously discussed or *two-sided*, in which case the alternatives obtained by exchanging X and Y are also considered (i.e., Y tending to be larger than X).

The test is based on counting the number of times a sample from one population precedes a sample from the other and thus requires an ordering of the samples.

Next we review some multivariate tests for the two-sample problem against any alternative. It is important to remark that concepts like cumulative distribution functions, runs or ranks do not have a straightforward generalization for the multivariate case.

Multivariate Two-sample Tests Methods

Next, we discuss some approaches to design two-sample tests in $\mathbb{R}^d$ and examples from the literature.

Extensions from One-dimensional Tests Although not straightforward, some extensions of 1D tests have been attempted in the literature.

For the Mann-Whitney test, the order can be obtained by looking at the sign of the differences between every pair of samples. Multivariate extensions of this test have been proposed in [MO95, OR04], by extending the sign concept to the *spatial sign* function defined as

$$\mathbf{S}(\mathbf{x}) \equiv \begin{cases} \|\mathbf{x}\|^{-1}\mathbf{x}, & \mathbf{x} \neq 0 \\ 0, & \mathbf{x} = 0 \end{cases} \tag{1.56}$$

and then considering the pooled sample $z^n \equiv x_1, \ldots, x_{n_0}, y_1, \ldots, y_{n_1}$ and the *spatial centered rank*

$$\mathbf{R}_i \equiv \frac{1}{n}\sum_{j=1}^{n} S(A_x(z_i - z_j)) \tag{1.57}$$

where A_x is a data based transformation chosen to make the test affine invariant. Then,

$$\bar{\mathbf{R}}_k \equiv \frac{1}{n_k}\sum_{j=1}^{n_k} \mathbf{R}_{(j+\sum_{l<k} n_l)} \tag{1.58}$$

is a measure of how much the k-th sample differs from the pooled one. The statistic is based on combining this measure for each population:

$$U^2 \equiv \frac{d}{c_x^2}\sum_{k=0}^{1} n_k \|\bar{\mathbf{R}}_k\|^2 \tag{1.59}$$

where $c_x^2 \equiv \frac{1}{n}\sum_{i=1}^{n} \mathbf{R}_i^T\mathbf{R}_i$. This test can be easily extended to the k-sample problem by extending the sum to k summands.

Notice that, contrarily to other tests based on interpoint distances that we will discuss below, this test depends only on the direction of the differences between points and, in this sense, it generalizes the Mann-Whitney test.

Some other generalization of this type of tests have been proposed in [LL04] (along with some scale tests), using the *data depth* concept which provides an ordering of the sample data with respect to a generalized median point (see, e.g., [LPS99, ZS00, Ser06]).

Friedman and Rafsky propose the minimal spanning tree (MST) of the pooled sample set (considering some distance between the points) as a multivariate generalization of the univariate sorted list. Therefore, one of their MST based test generalizes the Wald-Wolfowitz runs test by defining a run as a subtree whose vertices belong to the same sample set. Therefore, the test statistic is the number of edges that connects points of X to points of Y. The computational cost of this method using Kruskal's algorithm for building the MST is $O\left((n_0+n_1)^2 \log(n_0+n_1)\right)$.

An attempt to generalize Kolmogorov-Smirnov test is proposed in [Pea83]. The basic idea is to consider the largest difference between the empirical cumulative distributions considering the 2^d ways of defining them.[4] For example, in 2D, given the usual coordinates x and y, we consider $P(<x,<y)$,$P(<x,>y)$, $P(>x,<y)$ and $P(>x,>y)$. Thus, it is necessary to look at every possible step of the empirical cumulative distributions, in every possible order. A step occurs when a point is encountered when cumulating along one of the coordinates. Thus, $(m+n)^d$ steps must be considered (corresponding to selecting one coordinate value from each possible point). This corresponds to analyzing each quadrant at each center that yields a different step: there are then $2^d(m+n)^d$ possible quadrants. This procedure is of exponential complexity in d and therefore has been essentially proposed for 2 dimensions. Some variations and improvements of this test have been proposed (see [FF87, LRH07]), but still, essentially, for 2 or 3 dimensions.

Divergence based Naturally, any statistic converging to a measure of dissimilarity as those discussed in Section 1.2.1 can be used for a two-sample test.

Two recent examples are the MMD test of [GBR+06, GFHS09, GBR+12] and the KL divergence based test of [PC08].

Projection/Scoring/Ordering based A general method to build two-sample tests is to map the multivariate to a one-dimensional space using some kind of projection, scoring or ordering. Then, if the first step does not depend on the populations but just on the pooled sample, one can apply a classical one-dimensional test on the obtained data.

For example, in [FR79], an ordering is obtained by some traversal of the minimum spanning tree of the pooled sample.

In [CAFR07], it is shown that, under suitable regularity conditions, just one randomly chosen one-dimensional projection determines a distribution on $\mathbb{R}^d$. More precisely, if $F_X = F_Y$, the distributions of the projected variables obviously coincide but, if $F_X \neq F_Y$, the probability that they coincide is zero. Thus, the proposed method consists in projecting the sample points in a one-dimensional space spanned by some random vector h and then applying the univariate Kolmogorov-Smirnov test to the projected points. Because of the random nature of the procedure, the authors suggest to repeat it with several projections in order to increase the power, since given a sample, not all the projections are equally good at revealing the differences, because of the natural loss of information incurred when projecting. When combining the results, it is necessary to take into account the multiple testing problem (i.e., the increased probability of getting false positives) by, for example, correcting it via the Bonferroni [Bon36] or the Benjamini-Hochberg-Yekutieli [BY01] procedures.

As remarked in [Fri04], any soft classifier (or, more generally, any scoring function) trained on a separate data set can be used to build a two-sample test, again, using a one-dimensional test on the probabilities (or scores). If the same set is used for training and evaluating, then the procedure is no longer valid but one can resort to a permutation method as will see later.

In this spirit, there is the Area Under the Curve (AUC) based test [CDV09] that splits the data into two sets. The first step consists in training a score function using the AUC maximization criterion on the first dataset. The second step consists in scoring the samples from the second dataset using the trained score function to finally apply a Mann-Whitney test.

[4] Note that, in fact, there are $2^d - 1$ independent orderings since one depends on all the others, for example, in 2D, $P(>x,>y) = 1 - [P(<x,<y) + P(<x,>y) + P(>x,<y)$.

Category based Another interesting approach suggested in [FR79], consists in dividing the pooled samples into two mutually exclusive categories, based on any criterion not involving the X or Y labels. Then a test like Fisher's [Fis22] can be applied to detect a tendency of one of the two sample set to be in one of the categories. For instance, Friedman and Rafsky proposed (for comparison purposes) to use the following criterion for defining the categories: based on the MST of the pooled sample, they count the number of nodes of degree 1 for each label.

Of course, two categories are far from enough to detect any kind of difference. Nevertheless, the procedure can be extended to more categories using the G-test [Kul59, KKK62].

Interpoint Distances based Now, we review some interpoint distance based tests.

In [MPB96], it is shown that, under some assumptions on the interpoint distance function $h(\cdot,\cdot)$, a multidimensional two-sample test can be reduced to one-dimensional interpoint distance testing, i.e., testing whether the distributions of $h(X,X')$, $h(X,Y)$ and $h(Y,Y')$ are equal.

The MMD based test [GBR+06, GFHS09, GBR+12] also consider interpoint similarities by means of a kernel function.

Another related test is [BF04], which considers the following statistic:

$$T_{n_0,n_1} \equiv \frac{n_0 n_1}{n_0+n_1}\left[\frac{1}{n_0 n_1}\sum_{j=1}^{n_0}\sum_{k=1}^{n_1}\|X_j - Y_k\| - \frac{1}{2n_0^2}\sum_{j=1}^{n_0}\sum_{k=1}^{n_0}\|X_j - X_k\| - \frac{1}{2n_1^2}\sum_{j=1}^{n_1}\sum_{k=1}^{n_1}\|Y_j - Y_k\|\right]. \quad (1.60)$$

Local Coincidences based There is a long tradition of defining two sample tests based on certain local coincidences, i.e., with some definition of neighborhood, these tests combine the number of samples from each population to determine the global level of mixture of the two populations.

For example, k-nearest-neighbor approaches have been proposed and analyzed in [FST73, Sch86, Hen88] . The statistic is usually defined as follows. Denoting by Z_i the pooled samples, the following is defined

$$I_i(r) \equiv \begin{cases} 1, \text{ if the } r\text{th nearest neighbor of } Z_i \text{ belongs to the same sample as } Z_i \\ 0, \text{ otherwise} \end{cases}. \quad (1.61)$$

Then, denoting $n \equiv n_0 + n_1$, the statistic is defined as

$$T_{k,n} \equiv \frac{1}{nk}\sum_{i=1}^{n}\sum_{r=1}^{k} I_i(r) \quad (1.62)$$

which is simply the proportion of all k nearest-neighbor comparisons in which a point and its neighbor are member of the same sample. Under the null hypothesis, a low value is expected, since samples should be well mixed.

Another notable member of this category is Rosenbaum's Cross-Match test [Ros05] whose construction allows an exact characterization of its null distribution, as we will discuss later. First, a *minimum distance non-bipartite matching* is computed over the pooled sample. Supposing that $n = n_0 + n_1$ is even[5], a non-bipartite matching divides the n points into $I = n/2$ pairs of two points in such a way as to minimize the total of the I distances within the $n/2$ pairs. Then, the test statistic is the number of pairs containing a sample from both X and Y. This procedure has a time complexity of $O\left((n_0+n_1)^3\right)$.

In [BG05], a partition of the space is performed and the statistic is the sum, over all the cells, of the absolute value of the difference of counts between the two populations (i.e., the L_1 distance).

Note that many of the KL divergence estimators presented in Section 1.3.3 fall in this category.

Null Distribution

Given some statistic that aims at revealing differences between distributions if any, the next step is to characterize its distribution under the null hypothesis which must be distribution free, since nothing is assumed about f_X and f_Y.

[5] the odd case is treated by using a pseudopoint $n+1$ having distance 0 to every point, then optimally matching the $n+1$ points and discarding the pair containing the pseudopoint

Exact Distributions In some cases, it is possible to give an exact formula for the null distribution. Therefore, in these cases, the empirical type I error, as the number of trials tend to infinity, will be close to the specified α level.

A common way to obtain such exact formulas is when the statistic is based on ranks (but not directly on the sample values) to consider all distinct label permutations $\binom{n_0+n_1}{n_0}$ for the given sample sizes. A classic example is the Mann-Whitney test where all possible permutations of labels are implicitly considered using a recursive formula. When the sample size is large it gets difficult to compute and one usually resorts to an asymptotic normal approximation.

Another example is Rosenbaum's Cross Match test [Ros05], where an exact formula is provided. Nevertheless, as for the Mann-Whitney test, it is difficult to compute for large sample sizes, therefore the author proposes an asymptotic approximation.

Note that there can be a mismatch between the empirical Type I error and the design level when ties are possible (i.e., when different permutations yield the same statistic value).

Non-asymptotic Bounds When exact distributions are difficult to derive, it is sometimes possible to find thresholds for which the type I error probability is bounded by α. In [GBR+12], several bounds are provided for the MMD, but, as the authors note, those are distribution-free bounds that hold even in the worst case scenario and therefore these tests are conservative by design.

Asymptotic Distributions Asymptotic distributions are usually easier to obtain than exact ones and many authors propose to use them as approximations. Note that this is different from approximating a known exact distribution. In [GBR+12], a test based on an asymptotic null distribution for an unbiased estimate of MMD is proposed, which yields a greater power than the tests based on bounds.

For the multivariate runs test proposed in [FR79] and further analyzed in [HP99], a normal distribution allows to characterize the asymptotic null distribution. Normal based asymptotic distributions are also proposed for the k-nearest-neighbors based tests [FST73, Sch86, Hen88] .

Bootstrapping Another approach is to approximate the unknown null distribution using the empirical distribution. The bootstrap method draws, with replacement, n samples from the empirical distribution defined by the pooled sample. Authors that use bootstrap generally prove that their tests are asymptotically of level α (see, e.g., [FLLRB12] for a discussion).

In [BF04], the asymptotic null distribution of the test statistic is derived and shown to be the limit of the bootstrap distribution. In [GBR+06, GBR+12], a consistent bootstrap method is also proposed to estimate the threshold.

Permutation Methods The permutation method (see, e.g., [Goo05]), also called without-replacement bootstrap, is an alternative approach to obtain a null distribution. It relies on the assumption that, under $\mathtt{H}_0$, samples are exchangeable. Instead of considering analytically all the possible permutations, permutation methods actually permute the labels and recompute the statistic each time.

Even for moderate sample sizes it can be impractical to consider all possible permutations. Therefore, authors usually propose to use a Monte-Carlo approach, i.e., to consider only a subset of all possible permutations.[6]

Contrary to the standard bootstrap, permutation methods with correct handling of ties (e.g., by breaking them at random) and proper formulas yield valid p-values and not approximations (see, e.g., [E+04, PS10]). More precisely, let m the number of permutations performed (these can be sampled with or without replacement from the space of permutations), then denote by b the number of times that the statistic yielded by a permutation is more extreme than the statistic yielded by the actual observed data, then a valid p-value is obtained by the following formula

$$p = \frac{b+1}{m+1}. \tag{1.63}$$

[6] Apparently, this was first introduced in [Dwa57]

When permutations are drawn without replacement and there are no ties, this formula gives an exact p-value, otherwise it provides an upper bound. In [PS10], an exact formula is proposed for the latter case.

This perfect control of the type I error probability is appealing. Nevertheless, it is usually not straightforward to prove consistency for permutation methods.

With respect to the number of permutations m, it has been shown in [Joc86] in the context of parametric tests, that there is a loss of power when using permutations but this loss decreases as m increases.

The permutation method is a quite general one that can be applied to any statistic. Many authors, in addition to other methods, include it in their works, for example [HT02, Fri04, AZ05, Sch86]. In [GFHS09], the implementation called MMD_{boot} uses a permutation method too.

Effect Size

The *effect size* is a general concept in statistics meaning a quantitative measure of the strength of a phenomenon, e.g.: the correlation between two variables, the regression coefficient, the mean difference, etc. See, e.g., [SF12] for a recent discussion on the practical importance of the effect size.

In the case of a two-sample comparison, if a significant difference is found, it might be slight or on the opposite, very large. When using real data, if large sample sizes are used, it is reasonable to expect to find significant differences due, for example, to some sampling variability. To know if an observed difference is not only statistically significant but also important (or meaningful), one needs to quantify the effect size.

For example, when considering restricted alternatives, like a difference in mean or variance, an estimate of the magnitude of the difference in these parameters is very important in addition to the statistical significance of the result.

The classical Mann-Whitney test can be complemented by the Hodges-Lehmann estimator Δ [HJL63], which estimates the median of the difference between a sample X and a sample Y by computing the median of the difference in the cartesian product of the values in the two samples. This quantity has the following critical property: if the values of one population are shifted by Δ, then, the difference between the two populations vanishes—i.e. there is no evidence to reject the null.

The multivariate extensions to the Mann-Whitney test presented in Section 1.4.2 propose analogues to the Hodges-Lehmann estimator based on the spatial centered rank (see, e.g., [Oja10]). Nevertheless, for more general multivariate differences, the question is much harder and has not been addressed in the literature except for some mentions like in [Fri04].

1.5 Contributions

The contributions of this thesis are organized in three chapters.

Chapter 2 In this chapter, we address the question that naturally comes after a two-sample test rejects the null hypothesis: why the two data sets do not look like coming from the same distribution? In other terms, we address the problem of expressing an effect size for the multivariate nonparametric two-sample problem. This chapter presents a two-stage method providing *feedback* on this difference.

The first step consists in estimating on each sample point a local discrepancy measure that is a decomposition of the Jensen-Shannon divergence. As discussed, in Section 1.2.1, this measure has several desirable properties when using $\theta_0 = \frac{1}{2}$, in particular, it is symmetric and bounded between $[0, 1]$. Moreover, it can be expressed as the expected value over $z \in \mathbb{R}^d$ with respect to the average density $\bar{f}$ of the KL divergence between the conditional distribution $\mathbb{P}(L = \cdot | Z = z)$ and the Bernoulli distribution with parameter $\theta_0 = \frac{1}{2}$. This KL divergence is thus taken as a measure of local discrepancy. The estimation of the conditional probability (i.e., a soft classifier) is done by means of a strongly pointwise consistent regressor based on k-nearest neighbors as explained in Sections 1.2.2 and 1.3.3.

In the second step, we use Morse theory which is concerned with the study of a height function h defined on a manifold. In particular, it studies the variation of the topology of sublevel sets, i.e., sets of the form $h^{-1}(]-\infty, \lambda])$ where λ is some level. The height function studied is defined by the aforementioned

discrepancy. Practically, we study a discretized version of this function using a nearest-neighbor graph connecting the sample points—the elevation of a sample being the estimated discrepancy. Only critical points of index 0 and 1 (corresponding, respectively, to minima and saddles) are considered, as in the Watershed Transform (used in image processing for segmentation). Then, Topological Persistence allows to identify persistent local minima of this height function (i.e., *basins* that last long before merging into others and thus represent more stable features) and to simplify it as clusters defining regions of points with high discrepancy and in spatial proximity.

Experiments are reported both on synthetic and real data (satellite images and handwritten digit images), ranging in dimension from $d = 2$ to $d = 784$, illustrating the ability of our method to localize discrepancies.

Chapter 3 In Section 1.2.2, we discussed how soft classification is intimately related to dissimilarity. In Section 1.4.2, we mentioned how the AUC based test trains a scoring function on a portion of the dataset and then scores the remaining samples using this function in order to obtain the final statistic.

In this chapter, instead of using the train/test paradigm, we propose a sequential procedure for nonparametric two-sample testing using online soft classifiers. More precisely, we show that any online soft classifier can be turned into a sequential two-sample test for which a valid p-value can be computed, yielding controlled type I error. Thanks to its sequential nature, our test benefits from the property of optional stopping meaning that the user can decide to stop at any time, always obtaining a valid p-value. This is in contrast to Wald's procedure (discussed in Section 1.4) that requires the rule to be fixed in advance and to the linear version of MMD [GBR+12] which, even if processing is done in a sequential way, requires to fix in advance the sample size at which the rejection decision will be evaluated. This makes our test the first truly sequential nonparametric two-sample test.

We also show how to build consistent tests and, in particular, how to build them from a k-nearest neighbors nonparametric regressor. We also show that Bayesian mixtures and switch distributions can be used to increase power, while keeping consistency.

Chapter 4 Following the lines of Chapter 3, we build a sequential two-sample test using a spatial partition based sequential soft classifier instead of a nearest neighbors based one. The soft classifier used is *semi-supervised* meaning that, in an initial step, it uses an unlabeled set (e.g., a mix from the two populations being compared) to build the spatial partitions.

More precisely, our approach is based on Bayesian and Switch mixtures on an ensemble of partition based soft classifiers created by means of random projection trees [DF08] (see Section 1.3.3). The partitions discretize the density functions that are being compared, to obtain corresponding probability mass functions.

If the chosen partitions reveal the differences between densities (i.e., probability mass functions are different), then the corresponding two-sample test is proved to be consistent.

The algorithm can be used in a streaming context since it has, per sample, a negligible memory footprint and logarithmic time complexity w.r.t. the unlabeled set size constant time complexity, which remains constant.

Importantly, this construction automatically adapts to the scale at which differences occur and does not require parameters to be set.

Chapter 2

Beyond Two-sample-tests: Localizing Data Discrepancies in High-dimensional Spaces

2.1 Introduction

2.1.1 Comparing datasets in high dimensional spaces

Datasets represented as point clouds are ubiquitous in science and engineering, used for applications in 2D and 3D space (e.g. to represent laser scans in mock-up design) as well as in high dimensional spaces (e.g. to represent images and documents, physical or biological phenomena, etc). In manipulating such data, several classes of questions are faced, such as matching, topological inference, or comparison. This latter endeavor, which is the topic of this work, calls for a discussion in three directions, namely statistics (two-sample tests), information theory and learning (divergence estimation), and geometry-topology.

From a statistical standpoint, a broad class of comparison methods, requiring tame assumptions on the data are nonparametric two-sample tests (TST), see, e.g., [GBR+12] and the references therein. In a nutshell, a TST is a statistical hypothesis test checking whether the two sets can be seen as i.i.d. samples of an identical unknown distribution (the null hypothesis). In accepting or rejecting the null hypothesis, under a level of statistical significance α, a TST summarizes the body of information encoded in the points' coordinates into a single boolean value [Fri04]. However, this boolean information is in general of limited interest, for several reasons. First, it is unlikely that two real life datasets come from exactly the same distribution. Since consistent TST detect any kind of difference of any size if enough samples are given, the rejection of the null is expected. Second, the magnitude (and the nature) of the differences, known as *effect size*, usually conveys more information than the mere presence of a difference. Therefore, a reject decision should be just a signal to examine further the data in order to understand. Developing a notion of effect size for non-parametric two-sample tests in high dimensions has not been explored yet, and is the goal of this work.

From an information theoretical and probabilistic standpoint, the comparison can be phrased as the problem of estimating the global Kullback-Leibler divergence between unknown distributions using the samples in hand. For example, a difference between images has been proposed [FRD11], by coupling a univariate Kullback-Leibler estimate (per pixel) and a decomposition of the discrepancy map thus defined using a watershed transform in image space. In a more general setting, there exist techniques to estimate this quantity that avoid density estimation in high dimensions (see, e.g., [WKV05, WKV09, PC08, SN07]) and that could be amenable to decomposition, so as to determine a contribution of individual points or of groups of points. Nevertheless, this divergence lacks important properties (symmetry and boundedness), which makes it more difficult to further process it.

Finally, the comparison can also be tackled from the geometric and topological perspectives. In geometric terms, one may compute some distance or matching (one-to-one, one-to-many, many-to-many)

between the data, see [DX13] and the references therein. While this procedure is informative for the two datasets in hand, the main difficulty consists in accommodating a probabilistic setting. In a more topological perspective, persistence theory [EH10], which aims at assessing the stability of topological features–generators of persistent homology groups, was recently used to compare *persistence landscapes* [Bub15]. Such comparisons are clearly important since oblivious to geometric transformations, but our focus is clearly on geometry dependent features.

2.1.2 Contributions

This work proposes, to the best of our knowledge, the first attempt to model the differences (the effect size, see above), between two datasets for which one has rejected the null hypothesis stipulating that they share the same underlying distribution. In a nutshell, we aim at clustering samples, based on two criteria, namely samples within a cluster should (i) contribute significantly to the difference between the two clouds, and (ii) form a connected region. We match these goals by two major steps. In the first stage, we model pointwise differences, which we call *discrepancies*, using the Jensen-Shannon divergence (JSD), which is symmetric and can be decomposed in terms of the conditional probability of belonging to one of the populations given a space location. This conditional probability can be naturally estimated using known techniques of non-parametric regression like the one based on k_n nearest neighbours, which possesses strong asymptotic guarantees of consistency. In the second stage, using a nearest neighbor graph defined over the samples, we study the height function defined by the estimated discrepancy, and design a clustering procedure based upon topological persistence. Finally, we summarize the information by a bar plot giving the profile of the discrepancy aiming at representing a multivariate nonparametric effect size.

This work is organized as follows. Sections 2.2 and 2.3 respectively present the two major steps. Section 2.4 presents the construction of the bar plot and other plots that are useful to visualize the different steps of the method. Finally, Section 2.5 present experiments.

2.2 Estimating the discrepancy between datasets

We aim at modeling the discrepancy between two datasets $x^{(n_0)} \equiv \{x_1, \ldots, x_{n_0}\}$ and $y^{(n_1)} \equiv \{y_1, \ldots, y_{n_1}\}$, in some fixed dimension Euclidean space $\mathbb{R}^d$. We view these data as coming from two unknown densities f_X and f_Y with corresponding cumulative distributions functions F_X and F_Y.

2.2.1 Jensen-Shannon divergence decomposition using conditional distributions

Let $D_{\mathrm{KL}}(f\|g)$ be the Kullback-Leibler divergence (KLD) between two densities f and g defined as

$$D_{\mathrm{KL}}(f\|g) \equiv \int_{\mathbb{R}^d} f(x) \log \frac{f(x)}{g(x)} dx \tag{2.1}$$

with the conventions $0 \log 0 = 0$ and $0 \log \frac{0}{0} = 0$.

The Jensen-Shannon divergence (JSD) defined in [Lin91], allows to symmetrize and smooth the KLD by taking the average KLD of f_X and f_Y to the average density $f \equiv {}^{(f_X+f_Y)}/_2$, that is:

$$JS(f_X\|f_Y) \equiv \frac{1}{2}\left(D_{\mathrm{KL}}(f_X\|f) + D_{\mathrm{KL}}(f_Y\|f)\right) \tag{2.2}$$

In addition to being symmetric, the JSD is bounded between 0 and 1 and its square root yields a metric. Note also that by taking the average, two random variables are implicity defined: a position variable Z with density $f_Z \equiv f$ and a binary label L that indicates from which original density (i.e. f_X or f_Y) an instance of Z is obtained. Formally, considering the alphabet $\mathcal{A} = \{0, 1\}$ and $X \sim F_X, Y \sim F_Y$, the following pair of random variables is defined:

$$(L, Z) = \begin{cases} (0, X) & \text{with prob. } \frac{1}{2} \\ (1, Y) & \text{with prob. } \frac{1}{2} \end{cases}. \tag{2.3}$$

In the sequel, we will consider the conditional and unconditional mass functions $P(l|z) = \mathbb{P}(L = l|Z = z)$ and $P(l) = \mathbb{P}(L = l) = \frac{1}{2}$ respectively, as well as the joint probability density $f_{L,Z}$. We will also use the notation f_l to denote f_X (resp. f_Y) when $l = 0$ (resp. $l = 1$). Before establishing lemma 2.1, a key property for our comparison problem, we recall the definition of the Kullback-Leibler divergence between two discrete distributions P and Q over $\mathcal{A}$:

$$D_{\mathrm{KL}}(P\|Q) \equiv \sum_{l\in\mathcal{A}} P(l)\log\frac{P(l)}{Q(l)}. \tag{2.4}$$

Lemma. 2.1. *One has:*

$$JS(f_X\|f_Y) = \int_{\mathbb{R}^d} f_Z(z) D_{\mathrm{KL}}(P(\cdot|z)\|P(\cdot))\,dz. \tag{2.5}$$

Proof of lemma 2.1. Recall that the JSD can be expressed as follows:

$$\begin{aligned} JS(f_X\|f_Y) &\equiv \frac{1}{2}\left(D_{\mathrm{KL}}(f_X\|f_Z) + D_{\mathrm{KL}}(f_Y\|f_Z)\right) \\ &= \frac{1}{2}\int_{\mathbb{R}^d} f_X(z)\log\frac{f_X(z)}{f_Z(z)}dz \\ &+ \frac{1}{2}\int_{\mathbb{R}^d} f_Y(z)\log\frac{f_Y(z)}{f_Z(z)}dz \end{aligned}$$

.

By linearity of integration and by Bayes' Theorem, and noting that the conditional densities $f(z|L=0) = f_X(z)$ and $f(z|L=1) = f_Y(z)$ we have then

$$\begin{aligned} JS(f_X\|f_Y) &= \int_{\mathbb{R}^d}\sum_l P(l) f(z|L=l)\log\frac{f(z|L=l)}{f_Z(z)}dz \\ &= \int_{\mathbb{R}^d}\sum_l P(l)\frac{f_{L,Z}(l,z)}{P(l)}\log\frac{f_{L,Z}(l,z)}{P(l)f_Z(z)}dz \\ &= \int_{\mathbb{R}^d} f_Z(z)\sum_l P(l|z)\log\frac{P(l|z)}{P(l)}dz \\ &\equiv \int_{\mathbb{R}^d} f_Z(z) D_{\mathrm{KL}}(P(\cdot|z)\|P(\cdot))\,dz. \end{aligned}$$

□

The previous lemma shows that the JSD can be seen as the average, over $z \in \mathbb{R}^d$, of the KLD between the conditional and unconditional distribution of labels. More formally, we define:

Definition. 2.1. *The discrepancy at location z is defined as the KL divergence:*

$$\delta(z) \equiv D_{\mathrm{KL}}(P(\cdot|z)\|P(\cdot)). \tag{2.6}$$

Note that $\delta(z)$ ranges between 0 and 1 and is 0 iff $f_X(z) = f_Y(z)$. Note also that since $P(l)$ is known but $P(l|z)$ is not, the problem we consider now is the one of estimating $P(l|z)$ at each given location z.

2.2.2 Conditional probability estimation via non-parametric regression

In order to estimate the conditional distributions, we can use random design[1] non-parametric regression.

[1] In contrast to fixed design regression, where [GK02, Sec. 1.9]: "one observes values of some function at some fixed (given) points with additive random errors, and wants to recover the true value of the function at these points."

Generic Framework

First, we define the basic concepts (see, e.g., [GK02] for more details).

Definition. 2.2. *Given a random vector (Z, R), where $Z \in \mathbb{R}^d$ and the response variable $R \in \mathbb{R}$, the* regression function *is defined as*

$$m(z) = \mathbb{E}\left[R|Z = z\right]. \tag{2.7}$$

In the regression problem, the goal is to build an estimator $m_n(z)$ of $m(z)$ using a set of n i.i.d. realizations of (Z, R).

With respect to the guarantees that are provided for regressors, usually, the L_2 risk or mean squared error is considered, i.e.,

$$\int_{\mathbb{R}^d} |m_n(z) - m(z)|^2 \mu(dz) \tag{2.8}$$

where μ denotes the distribution of Z. Nevertheless, since our goal is to estimate the discrepancy $\delta(z)$, we seek pointwise guarantees for regressors. In particular, we will consider a strong form of consistency which is defined as follows.

Definition. 2.3. *Denoting μ the distribution of Z, a sequence of regression function estimates $\{m_n\}$ is* strongly pointwise consistent (s.p.c.) *if for μ-almost all $z \in \mathbb{R}^d$*

$$m_n(z) \xrightarrow{n\to\infty} m(z) \ \textit{a.s.}. \tag{2.9}$$

In [GK02, Sec. 25.6], some s.p.c. regression estimates are presented. For example, regression estimates based on partitioning, kernel and nearest neighbors are s.p.c under certain conditions for their parameters, when the absolute value $|R| < M$, for some M.

Now we describe the s.p.c. k_n-nearest neighbor regression function estimate (see [GK02, Ch.6&25] for further details). Given the training data $\{Z_i, R_i\}_{i=1,\dots,n}$, let us denote as $R_{(i,n)}(z)$ the response value corresponding to i-th nearest neighbor (with some tie-breaking rule) of z in $Z^{(n)}$. Then, the k_n-nearest neighbor (k_n-NN) regression function estimate is defined by

$$m_n(z) = \frac{1}{k_n} \sum_{i=1}^{k_n} R_{(i,n)}(z). \tag{2.10}$$

Then we have the following theorem [GK02, Thm. 25.17]:

Theorem. 2.1 (Strong pointwise consistency of k-NN)**.** *If $|R| < C$ for some $C < \infty$,*

$$\frac{k_n}{\log n} \to \infty \ \textit{and} \ \frac{k_n}{n} \to 0,$$

then the k_n-NN estimate using Euclidean distance is strongly pointwise consistent.

Application to Conditional Probability and Discrepancy Estimation

In order to apply this framework to our problem, note that the correspondence $R = L$ yields

$$m(z) = P(1|z). \tag{2.11}$$

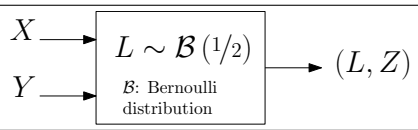
Then, we can use the following estimator for $P(l|z)$

$$\hat{P}_n\left(l|z\right) \equiv |1 - l - m_n(z)|. \tag{2.12}$$

Note that it is required that $0 \leq m_n(z) \leq 1$, since we aim at estimating a conditional probability, and that it is satisfied by the k_n-nearest neighbor regressor.
Using Eq. (2.12), we finally obtain an estimator for $\delta(z)$:

$$\hat{\delta}_n(z) \equiv D_{\mathrm{KL}}\left(\hat{P}_n\left(\cdot|z\right) \| P(\cdot)\right). \tag{2.13}$$

Figure 2.1 Random multiplexer generating pairs (label, position).



Theorem. 2.2 (Consistency)**.** *Let $\hat{P}_n$ be based on a s.p.c. sequence of regression estimates for (L, Z). Then,*

$$\hat{\delta}_n(z) \xrightarrow{n\to\infty} \delta(z) \text{ a.s.}$$

for f-almost all $z \in \mathbb{R}^d$.

Proof. Let us write $\hat{\delta}_n(z) = h(m_n(z))$, with $h(x) \equiv x \log(x/(1/2)) + (1-x)\log((1-x)/(1/2))$. It is easy to see that $h(x)$ is a continuous function (composition, sum and product of continuous functions). Then, we apply the continuous mapping theorem [Bil13], on every $z \in \mathbb{R}^d$ where $m_n(z) \xrightarrow{n\to\infty} m(z)$ to complete the proof. □

Remark 2.1. *Theorem 2.1, leaves many possible choices for k_n. For example, in [Gyo81, CD14], convergence rates of nearest neighbor based regressors and classifiers are studied. Roughly speaking, when $m(z) = P(1|z)$ satisfies an α-Holder like condition, it is shown optimal convergence rates, assessed by upper and lower bounds on classification rates, are obtained when $k_n = \frac{2\alpha}{2\alpha+d}$.*

2.2.3 Joint distribution compatible sampling

In the regression framework, the samples must be i.i.d. from a joint distribution $f_{L,Z}$. In our original problem, we have two sets of samples drawn independently from f_X and f_Y. In order to ensure this condition, we will use the random multiplexer depicted in Fig. 2.1. On each input it receives i.i.d. samples from each of the populations. Then, it generates an instance l of L. According to the value of l (0 or 1), it consumes the corresponding input (X, Y resp.) and outputs it along with l.

The following lemma shows that the output has the desired joint density.

Lemma. 2.2. *An output pair from the random multiplexer has joint density $f_{L,Z}$.*

Proof. The joint density of an output pair (l, z) is

$$\widetilde{f}(l, z) = \widetilde{f}(z|L = l)P(l).$$

Since, by construction, $\widetilde{f}(z|L = l) = f_l(z)$ and, by Eq. (2.3), $f_l(z) = f(z|L = l)$, then $\widetilde{f}(l, z) = f(z|L = l)\,P(l) = f_{L,Z}(l, z)$. □

Remark 2.2. *In practice, there is a finite set of i.i.d. samples of X and Y available. Then, at some point the multiplexer can have no more data to consume on one of the inputs while there is still data available on the other one. Therefore, some samples of the original sets would not be used and, thus, some loss of information is to be expected. To favor the inspection of all samples, we resample B times as follows:*

1. *For $j \in 1..B$ do:*
   (a) *Generate a sequence $\{z_{j_i}, l_{j_i}\}_{i=1,\ldots,m_j}$ using the random multiplexer, until one of the datasets is exhausted. Let m_j be the number of samples collected.*
   (b) *Build $\hat{\delta}^j_{m_j}(z)$ using some consistent estimator trained on $\{z_{j_i}, l_{j_i}\}_{i=1,\ldots,m_j}$*
2. *Define $\hat{\delta}(z) \equiv median_{j\in 1..B}(\hat{\delta}^j_{m_j}(z))$*

Notice that $\hat{\delta}(z)$ is also a consistent estimator.

Remark 2.3. *Although a random multiplexer that uses a Bernoulli distribution with parameter $\theta_0 = n_0/n$ instead of $1/2$ would exempt us from using the resampling procedure, this would require considering the Jensen-Shannon divergence with arbitrary weights (see [Lin91, Sec. IV]). The resulting effect size would lack the symmetry and normalization properties, therefore making the comparison between different effect sizes harder.*

2.3 Localizing the discrepancy

2.3.1 Defining clusters from sublevel sets

Overview. We wish to identify groups of samples, called *clusters*, intuitively characterized by two properties (Fig. 2.2):

- first, the discrepancy of such samples should be significant;
- second, samples within a cluster should be associated with regions where the discrepancy peaks.

To meet the first goal, we assume the existence of a value δ_{max} stipulating that below δ_{max}, the discrepancy is not significant. As we shall see in Experiments, while this value is not unique in general, few *clusters* typically stand out, imposing a lower bound on δ_{max}.

To meet the second goal, we use a strategy reminiscent of *mode clustering* [CM02, CadOS11], a general clustering strategy consisting of defining a cluster from the catchment basin of a local maximum of a density estimate, as well as watershed transforms used in image processing [RM00]. Our approach uses the hyper-surface defined by the discrepancy $-\delta(z)$ (Eq. (2.6)). In the sequel, we refer to this hyper-surface as the *landscape* of the *height function*, instinctively. In studying this landscape, clusters shall be defined from the *catchment* basins of selected local minima of the landscape.

In the sequel, we present the required background in the continuous setting, and further explain how to deal with the discrete datasets at hand.

Background. As stated above, we wish to focus on samples of the landscape whose discrepancy is less than $-\delta_{max}$ (Fig. 2.2). The region of the landscape matching this specification is called the sublevel set associated with $-\delta_{max}$. We define a cluster as a connected component (c.c.) of this sublevel set. We also note that each such cluster may be associated with the lowest local minimum of this connected component. However, in case of noisy estimates, many small clusters may be reported. To ease the interpretation of results, we therefore smooth the landscape, and focus on c.c. of the smoothed landscape.

The smoothing is carried out using *topological persistence* applied to the height function $-\delta(z)$. To explain this process intuitively, we borrow concepts from Morse theory [FK97]. The central theorem from Morse theory tells us that the topology of sublevel sets of a smooth function (of a Morse function, to be exact) changes upon passing critical points of the height function [Mil63]. For example, imagine sweeping the landscape with a horizontal hyper-plane, in a bottom-up fashion: when the plane hits a local minimum, a c.c. of the sublevel set defined by the height of the minimum is created; this c.c. initially reduces to the minimum, and further grows to its catchment basin.

Since our focus is on c.c. of sublevel sets, two types of critical points matter. The first type consists of local minima, as each local minimum creates a c.c. of a sublevel set; we define the *birth date* of that c.c. as the height of the local minimum. The second type consists of index one saddles, since such a critical point joins the c.c. of (generically) two local minima, or creates a loop (a generator in the so-called order one homology). For a saddle joining two c.c., the height of that saddle defines the *death date* of the c.c. which was born with the highest local minimum of the pair—recall that a saddle generically links two local minima. Using the previous construction, one assigns a birth date and a death to connected components of sublevel sets, except that associated with the global minimum—which never dies. Representing these points in 2D, with the x and y axis respectively coding the birth and death dates, yields the so-called persistence diagram (PD) of connected components of sublevel sets [EH10]. In particular, the *persistence* of a local minimum is the difference between the death date and the birth date of its catchment basin, or

equivalently, the height difference to the lowest saddle connecting this local minimum to a deeper local minimum.

Equipped with these notions, smoothing the landscape consists of retaining local minima whose persistence is above a user defined threshold ρ.

Algorithm. The significance threshold δ_{max} and the persistence threshold ρ just discussed yield a partition of the PD into five regions defined by three lines (Fig. 2.3). A local minimum m of the landscape (and its catchment basin) actually gets qualified with respect to three criteria:

- Selected/rejected: m is selected provided that its birth date occurs before $-\delta_{max}$.
- Persistent/canceled: m is persistent if its persistence is $\geq \rho$, a user defined threshold.
- Filtered/un-filtered: the catchment basin of m is filtered if the death date of m is larger than the threshold $-\delta_{max}$. Note that this step removes the samples whose discrepancy is not significant–with respect to the δ_{max} threshold.

The possible combinations, illustrated on Fig. 2.3, are:

- $m \in R_1$: rejected. Such a local minimum is rejected, since its discrepancy is less than δ_{max}. No point of the catchment basin of m is found in a cluster reported.
- $m \in R_2$: selected / canceled / un-filtered. Such a local minimum is selected, yet canceled by persistence. Because m dies before $-\delta_{max}$, all samples found in its catchment basin shall be part of a cluster reported.
- $m \in R_3$: selected / canceled / filtered. A local minimum which is selected, yet canceled by persistence. However, because m dies after $-\delta_{max}$, only the portion of its catchment basin belonging to the sublevel set $D_{\leq -\delta_{max}}$ shall be found in a cluster reported.
- $m \in R_4$: selected / persistent / un-filtered. Such a local minimum is selected, and is not canceled by persistence. Because m dies before $-\delta_{max}$, all the samples found in its catchment basin are found in a cluster reported.
- $m \in R_5$: selected / persistent / filtered. This combination is similar to the previous case, except that samples whose discrepancy is less than δ_{max} are discarded. Note that the cluster associated with the global minimum belongs to this region even though it is not found on the PD since the global minimum never dies.

From the previous discussion, one gets the following:

Observation. 1. *The number of clusters reported is equal one plus the number of points found in the region R_5 of the persistence diagram. These clusters are contributed by a number of persistent local minima equal to one plus the number of points found in the region $R_4 \cup R_5$ of the persistence diagram.*

This observation also has the following practical implication: for a fixed number of clusters, increasing (resp. decreasing) the threshold δ_{max} results in smaller (resp. larger) clusters, involving samples with high (resp. low) discrepancy.

Figure 2.2 Localizing the discrepancy: algorithm illustrated on a toy 1D example. (A, discrepancy estimation) We consider the height function defined by the discrepancy $-\delta(z)$, called the landscape for short. Practically, the estimate $\hat{\delta}(z_i)$ is used at each sample z_i (see text for details). **(B, clusters as connected components)** Focusing on discrepancies larger than a user defined threshold δ_{max}, clusters are defined as connected components (c.c.) of the sublevel set of the landscape defined by δ_{max}. On this example, four such c.c. are observed, two large ones and two small ones. Note that each cluster is charged to one local minimum of the function $-\delta(z)$. **(C, smoothed clusters upon applying topological persistence)** To get rid of small clusters, the landscape is smoothed using *topological persistence* (see text for details). In our case, this simplification yields two clusters. Note that the blue sample squeezed between C_1 and C_2 does not appear in the red cluster since its discrepancy is below δ_{max} (and likewise for the red point with respect to C_3 and C_4).

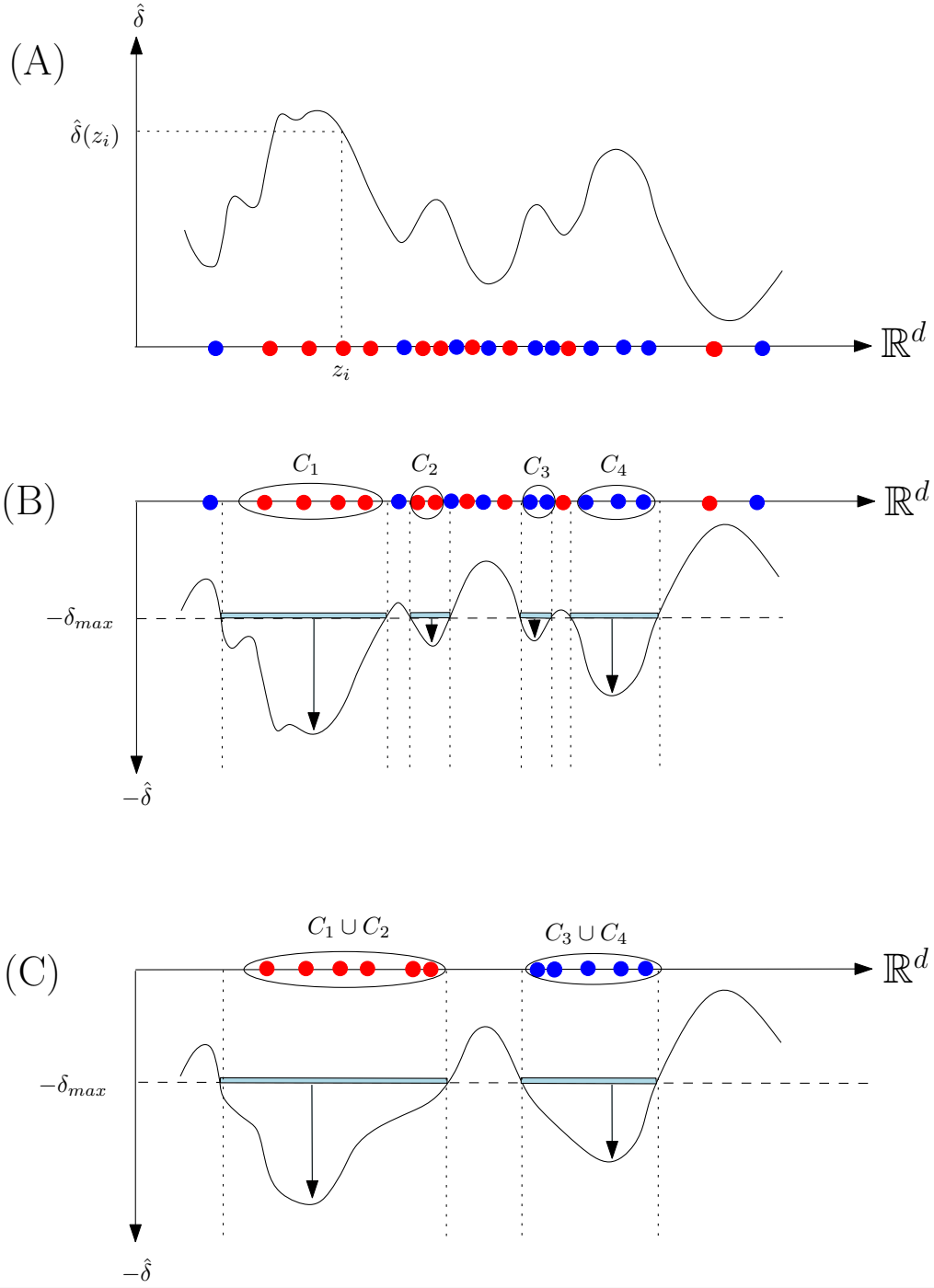
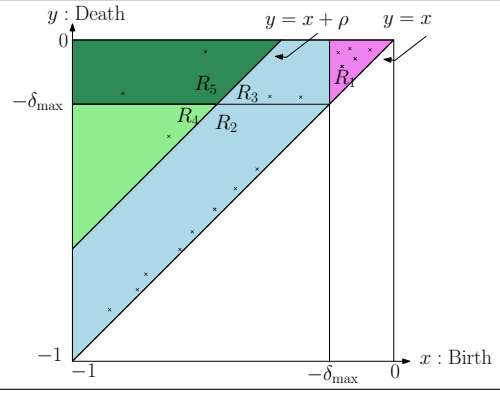


Figure 2.3 Partition of the persistence diagram exploited to define clusters. The axes x and y correspond respectively to the birth and death dates. The partition of the domain $y \geq x$ is induced by three lines: (i) $y = x + \rho$ which specifies the persistence threshold (ii,iii) $x = -\delta_{max}, y = -\delta_{max}$, with δ_{max} the threshold on the significance of the discrepancy. See text for the specification of regions R_1 to R_5.



2.3.2 Data structures and algorithms

Having presented the clustering algorithm for a smooth landscape, we now transcribe the method to handle the sample points z_i, each equipped with an estimate $\hat{\delta}(z_i)$. Note that the following points need to be addressed:

- data structures to represent the landscape,
- identification of critical points from the sampling,
- computation of the catchment basins of the local minima,
- implementation of the persistence based simplification.

An elementary strategy. To represent the discretization of the landscape provided by the sample points, we build a nearest neighbor graph (NNG) connecting the samples, that is, we connect each sample to its k nearest neighbors–using for instance the Euclidean distance. Second, we lift this graph into $\mathbb{R}^{d+1}$ by adding a new coordinate to each sample, namely $\hat{\delta}(z)$.

The NNG representation provides a simple way to study our estimated discrepancy [CadOS11]. To do so, let the star of a sample point be the set of its neighbors in the NNG. A local minimum is a sample such that all samples in its star have a higher elevation. For a sample which is not a local minimum, say p, one may estimate the negative gradient of the height function studied using the sample q maximizing the slope from p to q. Note that starting from a sample which is not a local minimum, the process of iteratively following this pseudo-gradient ends up at a local minimum. This process is called the *flow process*, and all samples flowing to a local minimum define its catchment basin.

Finally, to identify index one saddles, we proceed as in [CadOS11], processing the samples by increasing elevation $\hat{\delta}(z)$. In doing so, a saddle is a sample flowing to a local minimum, but having a neighbor in its lower star flowing to a distinct local minimum. (If several samples meet this criterion for the same two local minima, that with the lowest elevation only is retained.) Note that this construction aims at capturing the merge event between smooth sublevel sets, as described in the previous section.

Upon detecting a saddle, the persistence of the highest local minimum involved is computed, and compared against the threshold ρ. In case this persistence is less than ρ, a merge between catchment basins is performed using Union-Find [Tar83]. Finally, the inspection of samples halts upon reaching the first sample whose elevation is above $-\delta_{max}$—or equivalently discrepancy below δ_{max}.

A refined strategy. A computational drawback of the previous method is to focus on samples, while the construction of clusters uses the two criteria δ_{max} and ρ. The latter criterion being inherently coupled to the persistence diagram, i.e. to the critical points rather than the non critical samples (Fig. 2.3), we now present a more efficient method fully exploiting critical points and their catchment basins. This refined algorithm is especially interesting when the number of samples and the number of critical points differ by several orders of magnitude, which is usually the case. The algorithm is more involved as it requires notions from Morse homology [BH04]. In the sequel, we therefore only sketch its main steps, referring the reader to the supplemental section 2.7.1 for more details. The main steps are:

- Step1: Pre-processing. The local minima and index one saddles are identified as explained above. The connections between local minima and saddle yielded by the aforementioned flow process are used to define a graph connecting critical points. This graph aims at reproducing the so-called Morse-Smale-Witten complex used in Morse homology [BH04]. The catchment basin of each local minima is also built and stored with this local minimum.

- Step 2: Landscape simplification. Using the MSW complex, the landscape is simplified using topological persistence, using the recursive procedure explained in [CCS11]. This simplification manipulates the MSW complex rather than the samples–since the catchment basin is stored with the local minimum, whence its efficiency. The output consists of local minima with persistence above the threshold ρ, together with merged catchment basins.

- Step 3: Extraction of clusters. Finally, the connected components of sublevel sets are extracted from the simplified landscape.

Remark 2.4. *The previous description focuses on the identification of samples with significant discrepancy. In studying the height function $\delta(z)$ instead of $-\delta(z)$, one can instead identify clusters of samples of low discrepancy.*

2.4 Combining discrepancy estimation and localization

2.4.1 Qualifying the clusters

We decompose the JSD by clusters of points that are defined by the method described in Section 2.3. Then, the contribution of a cluster C reads as:

$$JS_C\left(f_X \| f_Y\right) \equiv \frac{1}{n_0+n_1} \sum_{z \in (x^{(n_0)} \cup y^{(n_1)}) \cap C} \hat{\delta}(z). \tag{2.14}$$

Combining the analysis of sections 2.2 and 2.3 yields the workflow of Fig. 2.4.

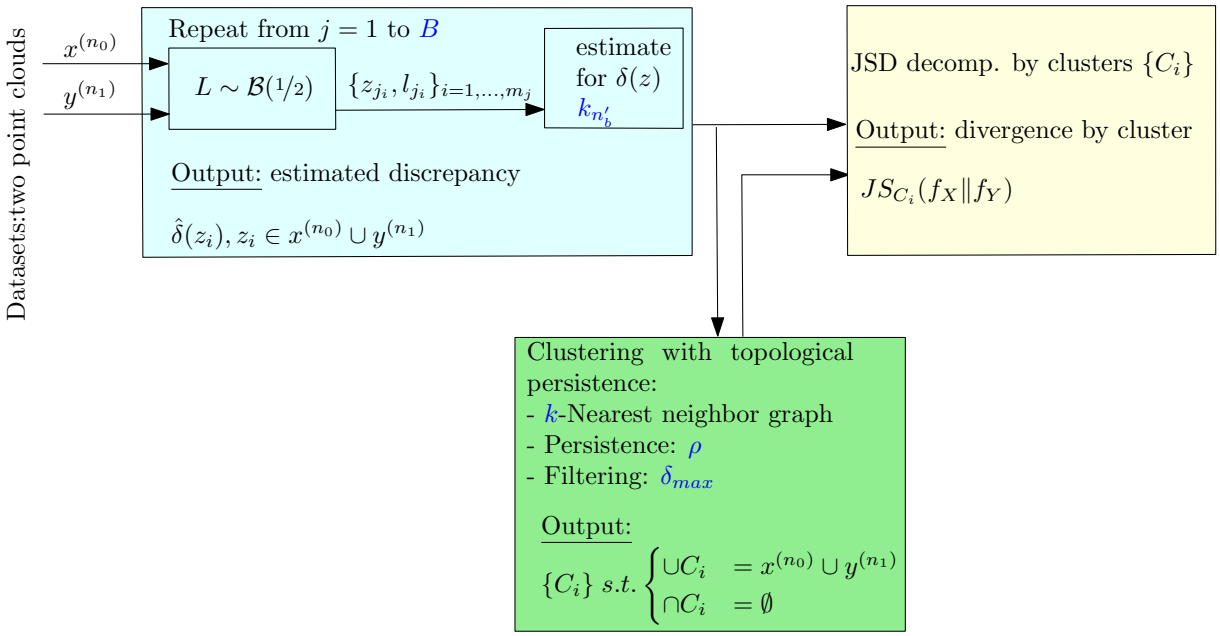
2.4.2 Plots

The previous analysis are best exploited using the following plots:

- *Raw data embedding:* For samples embedded in 2D or 3D space, a plot of the points with a color to indicate the label (blue: 0; red: 1). For samples embedded in a higher dimensional space, a 2D embedding of these samples obtained using multi-dimensional scaling (MDS). In any case, the goal of this plot is to intuitively visualize the distributions of the two populations.

- *Discrepancy shaded data embedding — aka discrepancy plot:* A plot similar to the raw data plot, except that each sample is color coded using a heat palette from fully transparent white to red across yellow, as a function of the value of the estimated discrepancy $\hat{\delta}(z)$. That is, a point with $\hat{\delta}(z)$ equal to zero (resp. one) is colored fully transparent white (resp. non transparent red).

Figure 2.4 Workflow of the whole method. In blue: the parameters.



- *Persistence diagram:* A plot showing for each minima a red cross with coordinates (x, y) corresponding to its birth and death dates respectively, while analyzing the landscape whose elevation is the negative estimated discrepancy.
- *Clusters*: A plot similar to the raw data plot, with one color per cluster. The points not belonging to any cluster are colored in gray.
- *JSD decomposition plot*: A 1D plot presenting a synthetic view without relying on the MDS embedding. The x coordinate represents the sample space and always ranges from 0 to 1. The total estimated JSD is represented by the area under the dashed line. And the maximum possible JSD which is always 1 is represented by the area under the continuous line. One bar is depicted for each cluster plus another one (the last one) for the points not belonging to any cluster. The area of each bar represents the contribution of the corresponding cluster to the total JSD and its color corresponds to the proportion of samples 0 in the cluster.

2.4.3 Implementation

Our method is implemented in the scope of the Structural Bioinformatics Library (SBL), available from `http://sbl.inria.fr/`. The SBL is a generic C++/Python library mixing low level algorithmic and data analysis classes, as well as applications targeting specific modeling problems in computational structural biology. Practically, our method is made available in `http://sbl.inria.fr/applications/` > Data analysis > Density_difference_based_clustering.

Practically, four python scripts are made available (see also the three steps in the workflow of Fig. 2.4):

- sbl-ddbc-step-1-discrepancy.py: the script estimating the discrepancy.
- sbl-ddbc-step-2-clustering.py: the script computing the clusters from the watershed transform.
- sbl-ddbc-step-3-cluster-plots.py: the script analyzing the clusters computed at the previous step.
- sbl-ddbc-disc-based-colored-embedding.py: an additional script computing an 2D embedding of the data processed, based on multi-dimensional scaling.

The user manual[2] precisely describes the parameters of these scripts as well as the formats of input and output files, making the results presented in this paper reproducible.

2.5 Experiments

We present results on four datasets featuring various difficulties: medium embedding dimension (real images), low intrinsic dimension (crenels), varying intensity of discrepancy (mixture of Gaussians), and real data embedded in high dimension (handwritten digits). For the sake of convenience, we refer to the two datasets to be compared as the blue and the red datasets. The number of neighbors was set to $k_n = n^{2/3}$. Note that the Maximum Mean Discrepancy two-sample test of [GBR+12] rejects the null hypothesis in all the cases for a significance level α lower than 1%.

2.5.1 Model: Gaussian mixture

Specification. The goal is to assess the ability of the method to spot regions of different intensity of discrepancy.

Two Gaussian mixture models where randomly generated using MixSim R package [MCM12] (see also supplemental section 2.8.1). The distributions for X and Y consist in two mixtures of four different two-dimensional Gaussians with equal weight and with some degree of overlap. In this example, $n_0 = n_1 = 2000$.

Results. The results are presented on Fig. 2.8 and 2.9 corresponding respectively to thresholds $\delta_{max_1} = 0.13$ and $\delta_{max_2} = 0.25$. The discrepancy plot clearly shows regions of different intensities. The persistence diagram (built with $k = 6$) highlights four persistent minima plus the global one, calling for simplification. On the other hand, the critical values associated with the local minima of the elevation are quite scattered. Setting $\delta_{max_1} = 0.13$ selects two large clusters, while setting $\delta_{max_1} = 0.25$ selects four smaller clusters, of which those identified by 0 and 2 exhibit higher discrepancies. This illustrates how δ_{max} trades-off the number of clusters, their discrepancy and their size.

In any case, note that the regions do not necessarily coincide with original components of the mixture, i.e., they are a result of the comparison only.

2.5.2 Model: crenels

Specification. The goal is to assess the ability of the method to spot local differences, and to cope with data of low intrinsic dimension (one) in a high dimensional space. We create two one dimensional datasets, which differ by two crenels.

More precisely, in this dataset, each point corresponds to a vector in $\mathbb{R}^{m\times m}$ encoding the pixels of a square grayscale image (0 = black, ≥ 1 = white) of size $m \times m$. The samples are the result of rotating the grayscale image i (Fig. 2.5). Therefore, taking $m = 11$ yields a dataset of intrinsic dimension one embedded in dimension $d = 121$.

Each sample of the blue population is an instance of a RV X obtained by rotating i with a random uniformly distributed angle A_X, that is:

$$X = rotate(i, A_X), A_X \sim \mathcal{U}(s, t). \tag{2.15}$$

where the function *rotate* applies a bilinear filter to smooth the result (details in [POSaH]).

For the red population, consider two Bernoulli random variables $B_1 \sim \mathcal{B}(p_1)$ and $B_2 \sim \mathcal{B}(p_2)$, and two uniform variables $U_1 \sim \mathcal{U}(a, b)$ and $U_1 \sim \mathcal{U}(c, d)$. Each sample of the red population is an instance of a RV Y defined as:

$$Y = rotate(i, A_Y), A_Y = B_1(B_2U_1 + (1 - B_2)U_2) + (1 - B_1)A_X. \tag{2.16}$$

Note that the rotation used to obtain Y comes from the uniform distribution $\mathcal{U}(s, t)$ with probability $1 - p_1$, and that the discrepancy between both distributions are high in the angle ranges $[a, b]$ and $[c, d]$

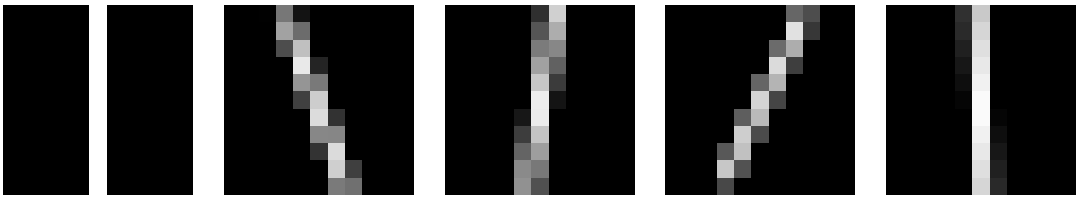
[2] `http://sbl.inria.fr/doc/Density_difference_based_clustering-user-manual.html`

(i.e. the *crenels*) and low everywhere else on the support, since, loosely speaking, points that are added to the crenels are missing from the rest of the support.

Practically, we used:

- $n_0 = 2000$, $n_1 = 2000$
- $s = -15, t = 15$
- $a = -4, b = -2, c = 9, d = 10$
- $p_1 = 0.3$, $p_2 = 0.5$

Figure 2.5 Rotated images (a) Original image i (b,c,d,e) Example rotated images



(a) (b) (c) (d) (e)

Results. Figure 2.10 shows the result of our method when applied to this dataset. The linear shape of the 2D MDS embedding illustrates the one dimensional nature of the data. The discrepancy plot hints at the crenels created by the two uniform distributions in Eq. (2.16). On the persistence diagram (obtained from a nearest neighbor graph with $k = 30$ nearest neighbors), we see a group of low discrepancy minima that are rejected by filtering out with $\delta_{max} = 0.1$ (dashed vertical line). One also identifies one persistent local minimum corresponding to the longer and, thus, the less red-concentrated crenel $[a, b]$. The other crenel corresponds to the global minimum which does not appear in the plot since its death date is infinite. The clusters yielded by the simplification and filtering steps correspond to the crenels. In the divergence decomposition plot, we observe these two crenels with high discrepancy produced by a high proportion of red points and also a non negligible total discrepancy given by the rest of the points, which has a higher proportion of blue points.

2.5.3 Model: mixture of handwritten digits

Specification. The goal is to assess the ability of the method to spot local differences, and to cope with real life high dimensional data.

This dataset is based on the MNIST dataset [LC98] that contains examples of handwritten digits (Fig. 2.6). Each digit is represented as a 28×28 gray-scale image, yielding a representation in dimension $d = 784$. For our experiment, we used the digits 3,6 and 8 and we built our populations by sampling with replacement from these three populations. The following table summarizes the number of samples taken from each digit set for each population:

digit	blue	red
3	100	1000
6	500	500
8	1000	100

Figure 2.6 Subsets of handwritten digits used. Images cropped from `http://www.cs.nyu.edu/~roweis/data.html`.

Results. The results are presented on Fig. 2.11. The persistence diagram (obtained from a nearest neighbor graph with $k = 30$ nearest neighbors) hints at one persistent local minima plus the global one, yielding two clusters corresponding to digits 3 and 8 as expected.

2.5.4 Statistical image comparison

When processing real images, two-sample tests are of mild interest since the null hypothesis is likely to be rejected. However, the JSD decomposition is still of interest to quantify the differences on a statistical basis.

Consider a digital image whose pixels use C color channels. For example, $C = 1$ in the monochrome case and $C = 3$ in the RGB color case (where the components of each vector correspond to the red/blue/-green intensities). A digital image is a $r \times c$ matrix of pixels, such that each pixel takes values in $[0, 1]^C$. (For a pixel, a value of 0 (resp. 1) represents the minimum (resp. the maximum) intensity for the corresponding color channel.) We follow the construction of [STS09] to build our samples, that is, by taking $b \times b$ pixel blocks yielding $(r - b + 1)(c - b + 1)$ samples, each being a vector of dimension Cb^2. Then, a discrepancy estimate $\hat{\delta}(z)$ is computed on each sample z and assigned to the pixel located in the upper left corner of the corresponding block. (NB: two bands of width $b - 1$ on the right and bottom side of the image are not assessed.)

Using the satellite color images of [STS09] shown in Figure 2.7 and using the same block size $b = 2$, we compute the estimated discrepancy $\hat{\delta}(z_i)$ for each sample z_i. The results are shown in Figure 2.12. The 2D MDS embedding is performed on a random subsample of size 4000 of the pooled dataset. The discrepancy plot hints at two clusters corresponding to colors found in the oval and the rectangular fields present in the second picture. These clusters stand out in the persistence diagram (obtained from a nearest neighbor graph with $k = 10$ nearest neighbors), the global minimum not being represented since its death date is infinite. We select them by setting $\delta_{max} = 0.1$.

In the JSD decomposition plot, we observe a small JSD (depicted by the dashed line) contributed by two relatively small clusters consisting of a strong majority of label 1. The local minima of the two clusters correspond to blocks in the second image as shown in Figure 2.13.

This experiment shows how our method can be used as a statistical image comparison tool, allowing to spot differences between two images by comparing densities in the color space. In this example, we expect the two green fields to be responsible of the statistical discrepancy and this is what we find in our analysis.

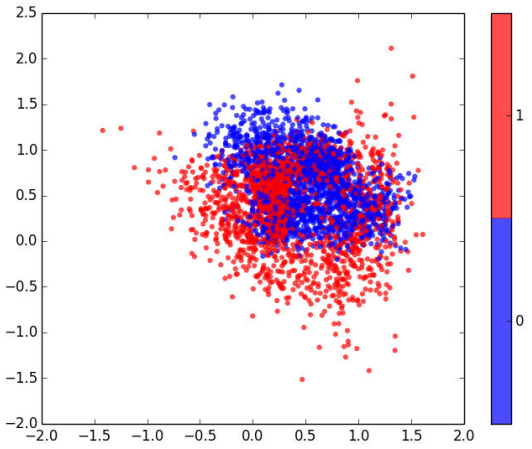
Figure 2.7 Original satellite color images from [STS09].

Figure 2.13 Original satellite image (Fig. 2.7): the red and blue squares correspond to the local minima of the two clusters identified from the analysis presented on Fig. 2.12.

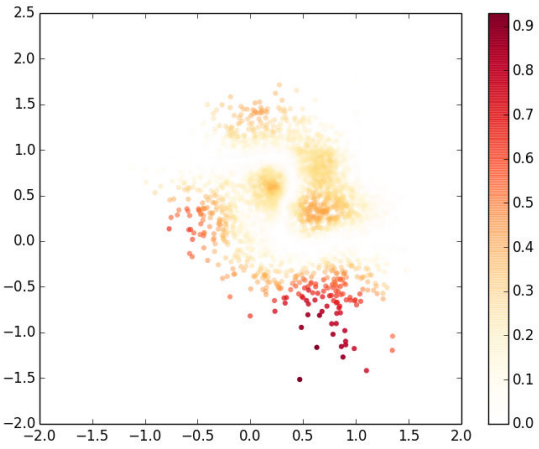
2.6 Conclusion

This work proposes the first method to model the difference between datasets given as point clouds, for which there is evidence showing that they do not have the same underlying distribution. The method relies on a pointwise estimation of an integrand related to the Jensen-Shannon divergence (JSD), a symmetric

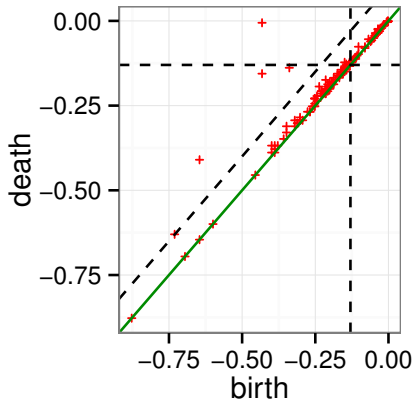
Figure 2.8 Model: Gaussian mixture. Figs, from top to bottom: Raw data embedding, Discrepancy plot, Persistence diagram, Clusters plot, Divergence decomposition



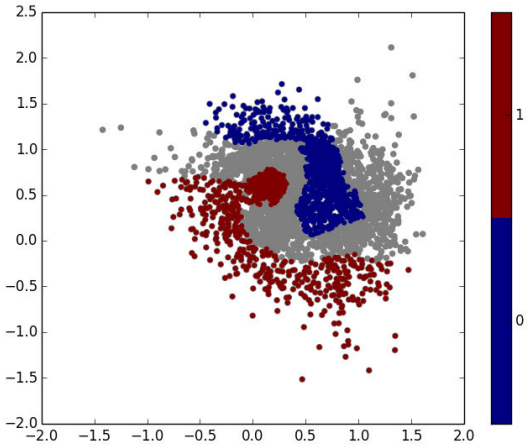
Raw data embedding



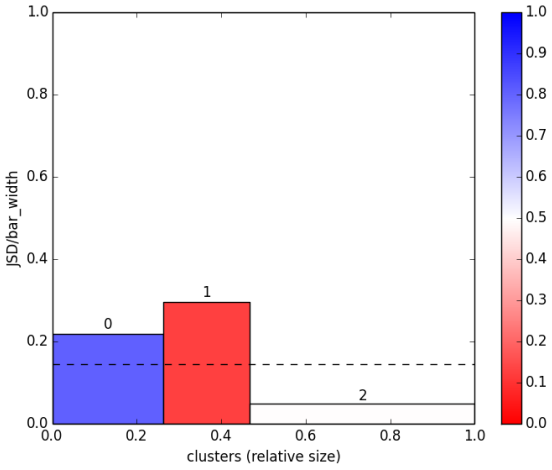
Discrepancy plot



Persistence diagram

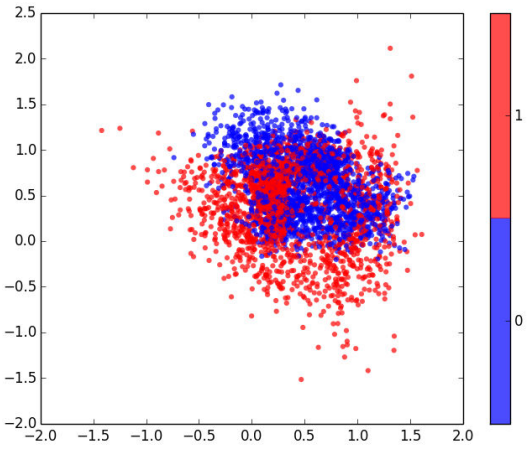


Clusters

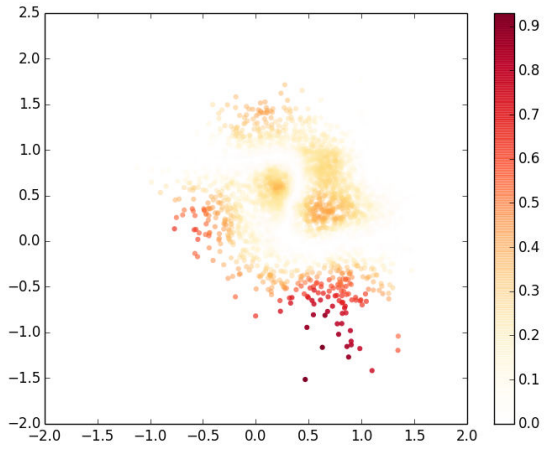


Divergence decomposition plot

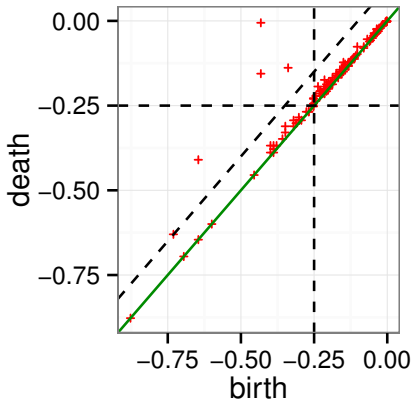
Figure 2.9 Model: Gaussian mixture. Figs, from top to bottom: Raw data embedding, Discrepancy plot, Persistence diagram, Clusters plot, Divergence decomposition



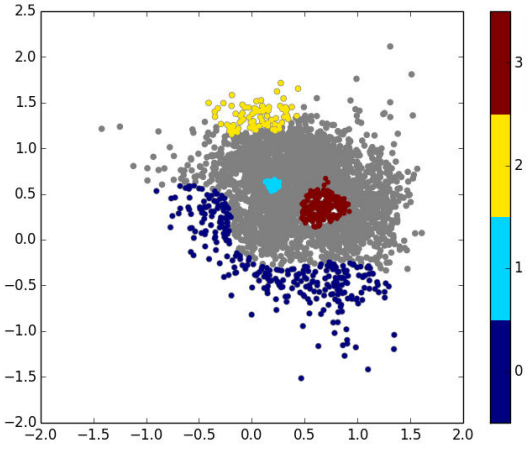
Raw data embedding



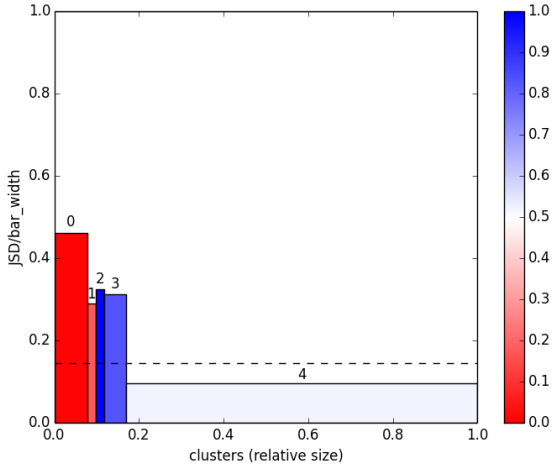
Discrepancy plot



Persistence diagram

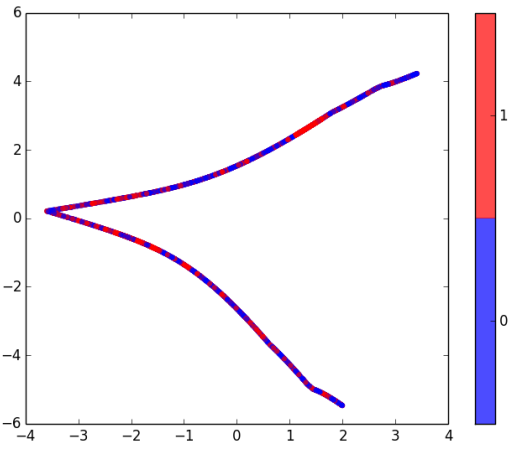


Clusters

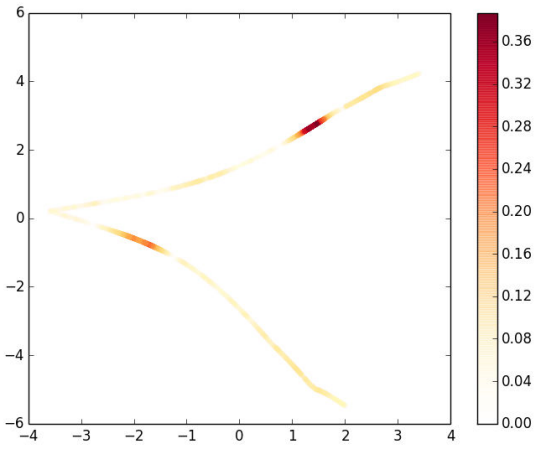


Divergence decomposition plot

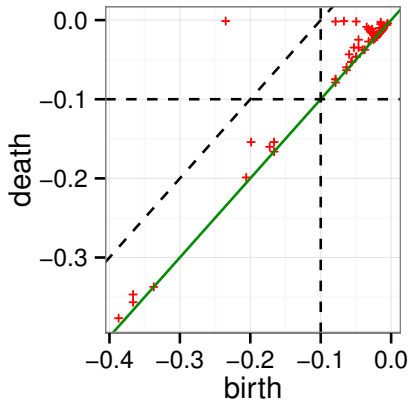
Figure 2.10 Model: Crenels. Figs, from top to bottom: Raw data embedding, Discrepancy plot, Persistence diagram, Clusters plot, Divergence decomposition



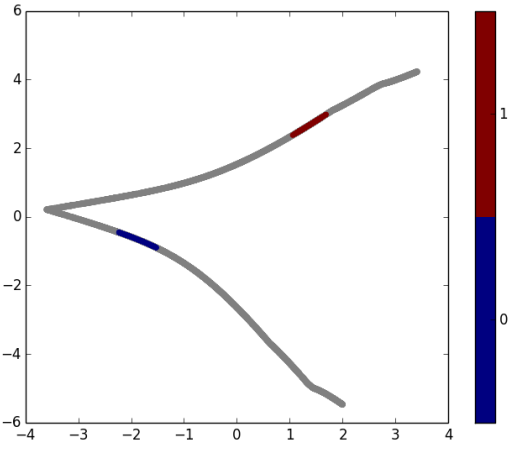
Raw data embedding



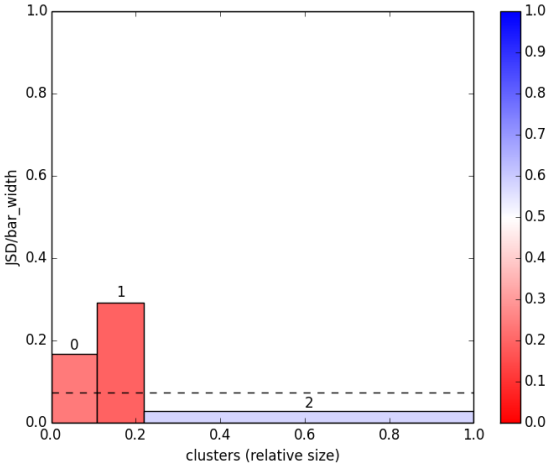
Discrepancy plot



Persistence diagram

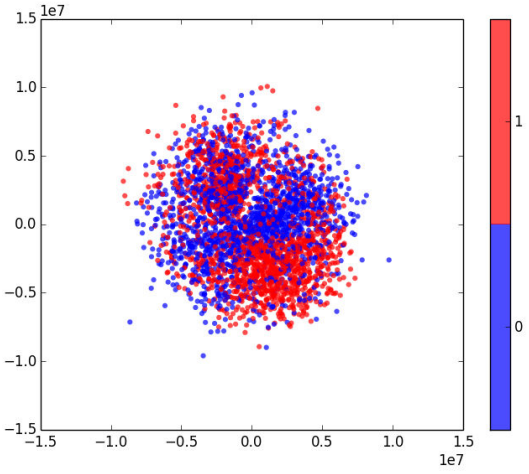


Clusters

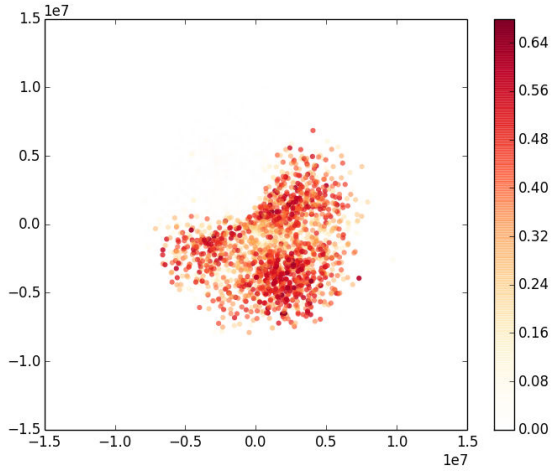


Divergence decomposition plot

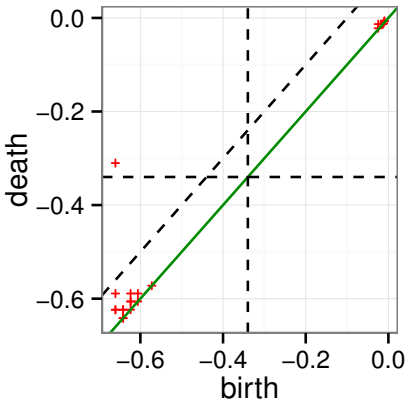
Figure 2.11 Model: Handwritten digits. Figs, from top to bottom: Raw data embedding, Discrepancy plot, Persistence diagram, Clusters plot, Divergence decomposition



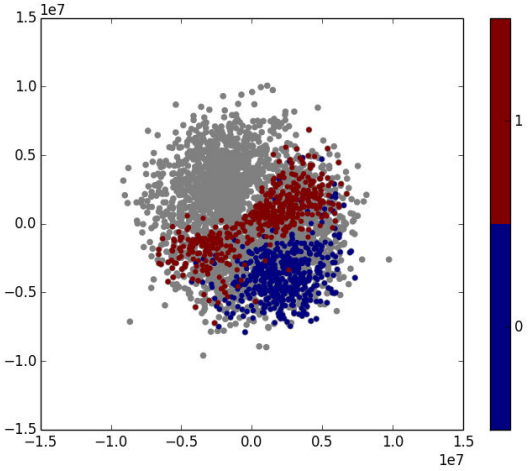
Raw data embedding



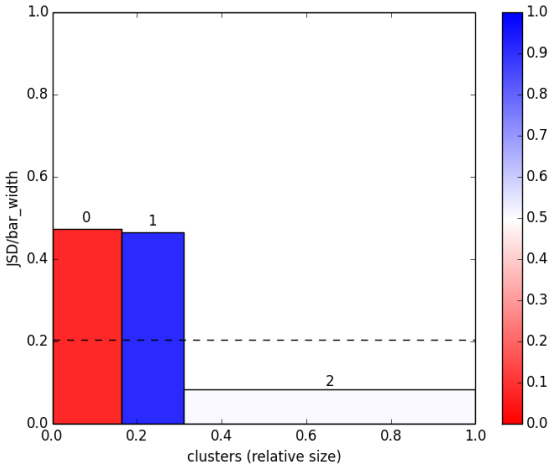
Discrepancy plot



Persistence diagram

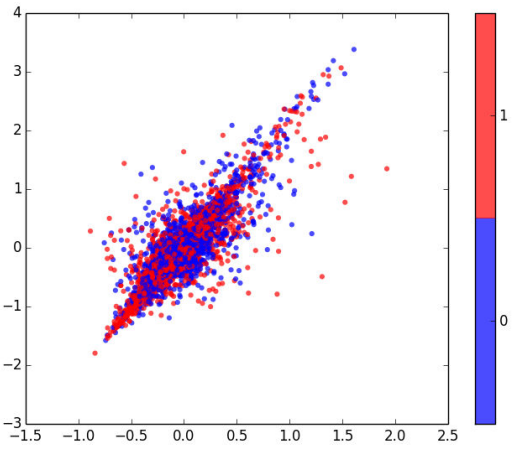


Clusters

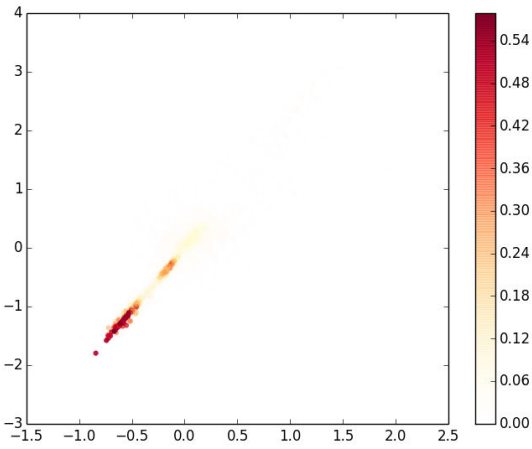


Divergence decomposition plot

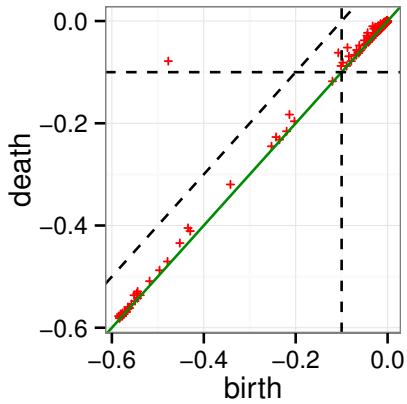
Figure 2.12 Model: Satellite images. Figs, from top to bottom: Raw data embedding, Discrepancy plot, Persistence diagram, Clusters plot, Divergence decomposition



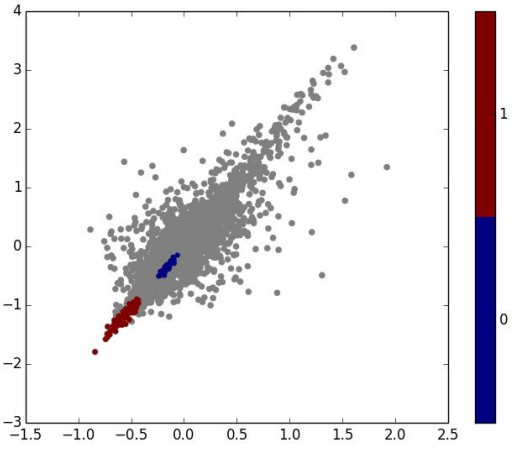
Raw data embedding



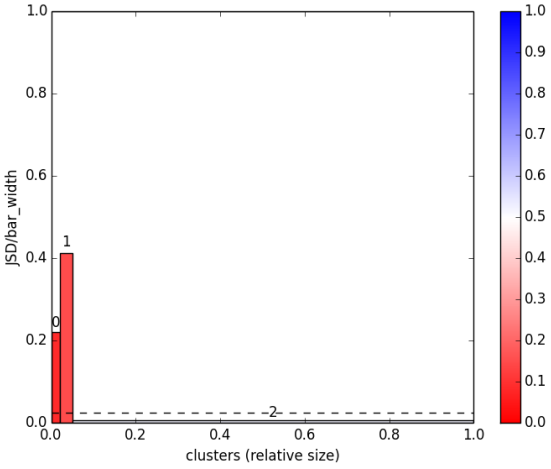
Discrepancy plot



Persistence diagram



Clusters



Divergence decomposition plot

version of the Kullback-Leibler divergence. An estimate of the JSD is obtained for each sample using a non-parametric regression method relying on k_n nearest neighbors estimates. Topological persistence is then used to gather samples in groups associated with local maxima of the JSD. All in all, our method delivers groups of samples with *significant* contribution to the JSD, and associated with local maxima of the JSD.

On the theoretical side, several questions are of major interest. A first goal will be to characterize the clusters returned by our procedure, based upon assumptions on the distributions underlying the data. This problem is related to the robustness of a recent clustering method combining mode seeking and topological persistence [CadOS11], since under suitable conditions, persistent modes of the density and those defining clusters have been shown to match. Coming up with a similar line of argumentation is more challenging in our case since two densities are involved, and the magnitude of the JSD and those of these densities are independent quantities. A second goal will consist of generalizing the method to data associated with a (non Euclidean) metric space.

On the practical side, we believe that our method goes well beyond statistical analysis based on two-sample tests, which essentially summarizes the information contained in all coordinates into a single boolean value (accept or reject the null hypothesis). It should therefore prove of interest wherever two-sample tests are used.

Acknowledgments. The authors wish to thank Tom Dreyfus for implementing the landscape analysis method used in Step 2.

2.7 Supplemental: Algorithms

2.7.1 A refined strategy to compute clusters

In this section, we provide further details on the refined strategy used to compute clusters, see section 2.3. Recall that the elementary strategy merely maintains the tree of merge events between catchment basins, also known as the disconnectivity graph, using a Union-Find algorithm. The refined strategy uses the Morse-Smale-Witten complex instead, namely the graph connecting critical points (Fig. 2.14).

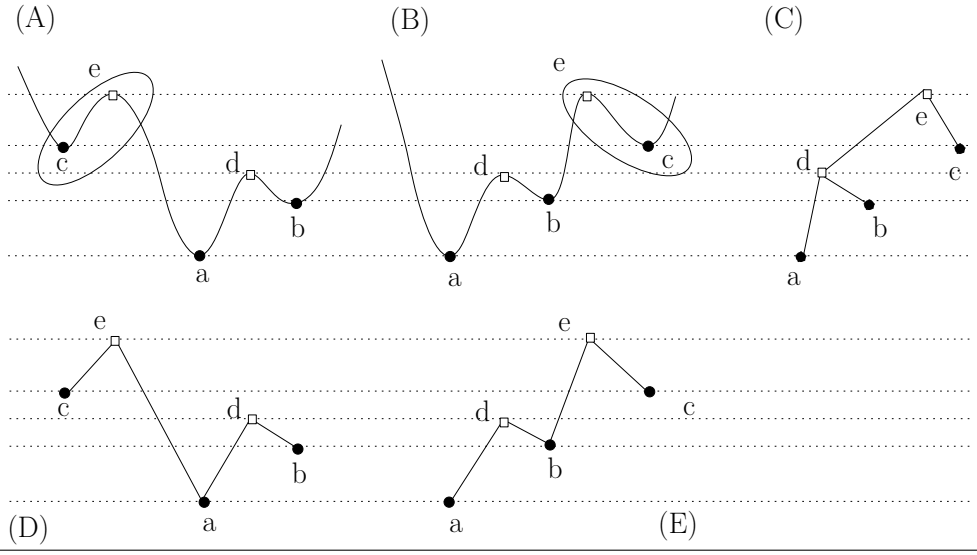
Step 1: construction of the MSW complex. From the critical points identified in the elementary strategy, we build the bipartite graph whose vertex sets are the local minima (index 0 critical points) and the saddles (index 1 critical points) respectively (Fig. 2.14(D,E)). We also build the stable manifolds of local minima (i.e. their catchment basins), and assign them to the local minima. Using this MSW complex, we also compute the persistence diagram associated with local minima.

Step 2: Simplifying the landscape. The general procedure to recursively simplify a landscape using the MSW complex has already been presented in the context of non manifold shape reconstruction [CCS11]. This procedure, which handles cancellation of non persistent critical points whatever their indices, is used here only for local minima and index one saddles. In a nutshell, the cancellation of a pair of critical points (a, b) whose indices differ by one consists of rerouting the connections of a and b in the MSW complex, and of redistributing the stable manifold of a [CCS11]. Note in particular that each remaining local minimum is endowed with two types of samples: those from its own SM and those from SM inherited from canceled local minima.

Step 3: sublevel set extraction. The previous simplification yields a partition of the landscape into the SM of the persistent minima. We remove from these SM the samples whose discrepancy is less than δ_{max}, a task carried out in two steps. First, the samples from its own SM are filtered out. Second, the samples of basins inherited from the simplification are also filtered out, provided that such a basin was born before $-\delta_{max}$. In particular, inherited basins born after $-\delta_{max}$ are ruled out in constant time. The persistent local minima and their remaining samples, if any, form the clusters.

Remark 2.5. *In the previous explanation, the NNG connecting samples has been taken for granted for the step one. The NNG construction must indeed be carried out first, as filtering out samples with low elevation may deplete the neighborhood of selected samples, and jeopardize the identification of critical points.*

Figure 2.14 Landscape simplification using topological persistence: simple strategy using Union-Find, versus refined strategy using the Morse-Smale-Witten (MSW) complex. (A,B) Two landscapes, with critical points of indices 0 (disks) and 1 (squares). **(C)** The disconnectivity graph (DG) of both landscapes, namely the tree depicting the evolution of connected components of sublevel sets. Despite the differences between their MSW complexes, both landscapes share the same DG: upon passing the critical point e, the stable manifold of c merges with that born at a. **(D,E)** The MSW complexes of **(A,B)**, respectively. In cancelling the pair of critical points (c, e), one does not know from the DG with which (a or b) the basin of c should be merged. But the required information is found in the MSW complex: on the landscape (A), c is merged with a; on the landscape (B), c is merged with b.



2.8 Supplemental: Data Sets

2.8.1 Gaussian mixture

Summary:

- $n_0 = n_1 = 2000$
- $d = 2$
- Rationale: A simple and easy to visualize dataset with regions of different intensities of divergence

The first population is drawn according to a mixture of four spherical Gaussians with equal probability, i.e.,

$$X = \sum_{i=1}^{4} 1_{\{i=U\}} N_0[i]$$

where U is a uniform discrete RV which takes values in $\{1..4\}$ and

$$N_0[i] \sim \mathcal{N}(\mu_0[i], \sigma_0[i])$$

The randomly generated parameters by MixSim R Package were:

$$\mu_0[1] = (0.9556715, 0.3617815)$$
$$\mu_0[2] = (0.6207539, 0.8498296)$$
$$\mu_0[3] = (0.3077166, 0.2886823)$$
$$\mu_0[4] = (0.1496965, 0.9773699)$$

$$\sigma_0[1] = 0.04771839 I_2,\ \sigma_0[2] = 0.02111844 I_2,\ \sigma_0[3] = 0.03342965 I_2,\ \sigma_0[4] = 0.07551961 I_2$$

where I_2 is the 2×2 identity matrix.

The second population is drawn according to a mixture of four non-spherical gaussians with equal probability defined by the following randomly generated parameters (using analogous notation):

$$\begin{aligned}
\mu_1[1] &= (0.00677118, 0.07022882) \\
\mu_1[2] &= (0.21864233, 0.46602229) \\
\mu_1[3] &= (0.99020950, 0.20540745) \\
\mu_1[4] &= (0.29765334, 0.80943535)
\end{aligned}$$

$$\sigma_1[1] = \begin{bmatrix} 0.1522406 & -0.1031709 \\ -0.1031709 & 0.1509098 \end{bmatrix} \sigma_1[2] = \begin{bmatrix} 0.006279164 & -0.001738228 \\ -0.001738228 & 0.032324880 \end{bmatrix}$$

$$\sigma_1[3] = \begin{bmatrix} 0.05161954 & 0.02275865 \\ 0.02275865 & 0.25600142 \end{bmatrix} \sigma_1[4] = \begin{bmatrix} 0.11359079 & 0.02248852 \\ 0.02248852 & 0.02819918 \end{bmatrix}$$

Chapter 3

A Sequential Non-parametric Two-Sample Test via Nearest Neighbors

3.1 Introduction

3.1.1 Background

Given two sets of samples $x_1, \ldots, x_{n_0}$ and $y_1, \ldots, y_{n_1}$ whose corresponding random variables $X_i \in \mathbb{R}^d$ and $Y_i \in \mathbb{R}^d$ are i.i.d. with densities f_X and f_Y respectively, a nonparametric two-sample test ambitions to determine whether $f_X = f_Y$. To this end, a *statistic*, i.e., a function aiming at revealing discrepancies of the data is defined. This function typically quantifies in a global way the *local homogeneity* of the mixture of the populations, this local homogeneity being assessed by methods as diverse as nearest-neighbors (e.g. [Sch86, Hen88, Ros05, PC08]), spatial partitions (e.g. [BG05]) or kernels (e.g. [GBR$^+$12]). The way the data are processed allows classifying two-sample tests into two tiers. Tests of the first tier compute a statistic on the whole dataset, to reveal the discrepancy if any (e.g. classical tests like [Bic69, FR79, Sch86, Hen88] and also more recent ones like [HT02, Ros05, GBR$^+$12]). Tests from the second tier use a learning perspective by splitting the data into a training and a test sets (e.g. [Fri04, CDV09, GSS$^+$12, ZGB13]). In the training phase, the function aiming at revealing the discrepancies is learned and optimized (in the sense of its discrimination power). In the second phase, this function is evaluated on the complement of the training set to obtain the statistic.

An appealing extension to this second tier is to use a sequential framework in which the function is optimized at each sample and then immediately used to test the next sample. In this sequential framework, it is natural and desirable to be able to stop at any time, i.e., as soon as a difference has been perceived. Whereas classical Neyman-Pearson null hypothesis testing requires the sampling plan or equivalently the stopping rule to be defined in advance to ensure the validity of the procedure (see, e.g., [Wag07]), Bayes factor model comparison makes this optional stopping possible (see, e.g., [SSVV11]). Note that selected tests, such as MMD_l [GBR$^+$12], process samples sequentially, yet require a predefined sampling plan.

3.1.2 Contributions

We make three contributions. First, we design a framework for sequential nonparametric two-sample tests. The framework is based on sequential prediction of labels defining the two populations (or equivalently distributions on length-n sequences), and enjoys optional stopping. Second, under suitable conditions qualifying the difference between the tested distributions, consistency of the two-sample test is guaranteed when the sequential predictor is built from strongly pointwise consistent regressors, obtained in our case from k_n-nearest neighbors (KNN) regressors. Third, we show that combing mixtures and switch

distributions is effective in increasing power, as our tests outperform state-of-the-art ones on selected challenging datasets.

We note in passing that our contribution bears two main differences with Wald's sequential test [Wal45]. First of all, this classical procedure works only for simple alternative hypotheses, since the probability of type II error must be kept under control. Although some extensions have been proposed (e.g., [SMFLPCAR08]), they are not applicable to the nonparametric case. Another important difference is that in Wald's procedure the stopping rule must be fixed in advance in order to obtain a valid p-value for the whole procedure, while in our procedure optional stop is allowed.

The previous contributions are articulated as follows. Lemma 3.1 connects classical two-sample testing to probabilistic classifiers. Theorem 3.1 and Lemma 3.2 turn any sequential predictor (and, thus, any non-anticipating probabilistic classifier) into a sequential two-sample test allowing optional stopping. Theorem 3.2 gives sufficient properties on probabilistic classifiers to guarantee λ-consistency (a weaker sense of consistency) of the resulting two-sample test. Theorem 3.3 shows how such conditions can be obtained from strongly pointwise regressors.

Since regressors are in general parameterized and may be optimized, in Section 3.4, we use Bayesian mixtures of regressors, promoting the regressors with better past performance. Additionally, we suggest to use switch distributions to attenuate the power loss that is expected due to sequential learning.

3.2 Two-sample test based on sequential prediction

3.2.1 Problem statement

We state the problem as follows:

Problem. 3.1. *Given a set of samples $x_1 \dots x_{n_0}$ and $y_1 \dots y_{n_1}$ whose corresponding random variables X_i and Y_i are i.i.d. with densities f_X and f_Y respectively, select one of the following hypotheses*

$$\begin{cases} \mathtt{H}_0 : f_X = f_Y \ \ a.e. \\ \mathtt{H}_1 : \neg \mathtt{H}_0 \end{cases} . \tag{3.1}$$

In order to assess the strength of the evidence against $\mathtt{H}_0$, a random variable p is used, which is called a *valid p-value* (see, e.g., [CB01, Def. 8.3.26]) if $0 \leq p \leq 1$ and

$$\mathbb{P}_{\mathtt{H}_0} (p \leq \alpha) \leq \alpha, \ \forall \alpha \in [0, 1]. \tag{3.2}$$

Then, the lower is p the stronger is the evidence against $\mathtt{H}_0$. It is also possible, to set a threshold called *significance level* α, so that $\mathtt{H}_0$ is rejected when $p \leq \alpha$.

Then, two types of errors must be considered. One faces a type I error when $\mathtt{H}_0$ is rejected while it is actually true. One faces a type II error when $\mathtt{H}_0$ is not rejected while it is actually false. The probability of Type I error is controlled by design and is upper bounded by α. Then, one usually considers the *power* of the test for a significance level α, which is $\mathbb{P}_{\mathtt{H}_1} (p \leq \alpha)$. The test is termed *consistent* for a given level α when $\mathbb{P}_{\mathtt{H}_1} (p \leq \alpha) \xrightarrow{n_0+n_1 \to \infty} 1$.

3.2.2 Random labels framework

A *sequential probability predictor* (or *predictor* for short) processes sequentially input symbols belonging to some alphabet $\mathcal{A}$. Before observing the next symbol in the sequence, it predicts it by estimating the probability of observing each symbol of the alphabet. Then, it observes the symbol and some loss is incurred depending on the estimated probability of the current symbol. Subsequently, it refines its model in order to better predict future symbols. The predictor can also be allowed to observe side-information to make better predictions.

Intuitively, if we shuffle the samples from both populations, and yet manage to predict the population each sample belongs to, then it is natural to think that there is some difference in the features, so that $\mathtt{H}_0$ should be rejected.

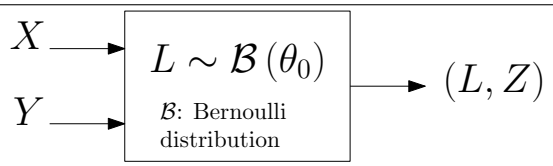
In order to do this shuffling, we use a *random device* receiving samples from each of the populations(Fig. 3.1). The output corresponds to the input X with probability θ_0 or to Y with probability $1-\theta_0$, where $0<\theta_0<1$ is a parameter that must be set. Formally, considering the alphabet $\mathcal{A}=\{0,1\}$, we define the following pair of random variables:

$$(L,Z)=\begin{cases}(0,X) & \text{with probability } \theta_0\\ (1,Y) & \text{with probability } 1-\theta_0\end{cases}.$$

In a classical two-sample test setting, the inputs of the random device uniformly draws from each of the given finite populations until the selected input has no more samples available, in which case the paired sequence generated $(L,Z)^N$ ends. Notice, that it can happen that $N<n_0+n_1$. In order to minimize the expected number of unused samples $N-(n_0+n_1)$, one should set $\theta_0=n_0/(n_0+n_1)$.

Remark 3.1. *If we are, for example, in a streaming setting where the inputs of the random multiplexer are plugged on streams with rates r_0 and r_1 (in symbols per time unit) it is natural to set $\theta_0=r_0/(r_0+r_1)$ in order to match them. In this case, the sequence never ends.*

Figure 3.1 The random multiplexer used to generate pairs (label, position)



3.2.3 Notations and problem reformulation

Our framework is based on considering two random variables Z (positions) and L (labels) representing, respectively, the pooled original samples, and the two populations these samples belong to. The following notations are used to describe the probabilistic properties of these random variables. The unconditional and conditional label probabilities are denoted $\mathbb{P}_{\theta_0}(l)$ and $\mathbb{P}_{\theta(z)}(l|z)$. One has:

$$\begin{cases}\mathbb{P}_{\theta_0}(l)\equiv\mathbb{P}(L=l),\\ \mathbb{P}_{\theta(z)}(l|z)\equiv\mathbb{P}(L=l|Z=z)\end{cases}\tag{3.3}$$

with $\theta_0\equiv\mathbb{P}(L=0)$ and $\theta(z)\equiv\mathbb{P}(L=0|Z=z)$. The joint density and the joint density assuming independence are respectively denoted $f_{\theta(z)}(z,l)$ and $f_{\theta_0}(z,l)$. The mixture density for position is denoted $f(z)=\sum_l f_{\theta(z)}(z,l)$.
The entropy of random variables is denoted $H(\cdot)$, while the entropy of L conditioned on Z is denoted $H(L|Z)$; finally, the mutual information between Z and L is denoted $I(Z;L)$.

Remark 3.2. *With the previous notations, one has*

$$f_{\theta(z)}(z,l)\equiv\mathbb{P}_{\theta(z)}(l|z)\,f(z),\ \textit{and}\ f_{\theta_0}(z,l)\equiv\mathbb{P}_{\theta_0}(l)\,f(z).\tag{3.4}$$

In the setting of random labels, let us consider the following two-sample problem:

Problem. 3.2. *Given a sequence of samples $(l,z)^n$ whose corresponding random variables (L_i,Z_i) are i.i.d. with joint density $f_{\theta(z)}(\cdot,\cdot)$, select one of the following hypotheses*

$$\begin{cases}\mathtt{H}_0 & :f_{\theta(z)}(z,l)=f_{\theta_0}(z,l)\ \textit{a.e.}\ \forall l\in\{0,1\}\\ \mathtt{H}_1 & :\neg\mathtt{H}_0\end{cases}.\tag{3.5}$$

The following lemma is a simple consequence of Bayes' formula applied to the joint densities.

Lemma. 3.1. *The null hypotheses of Problems 3.1 and 3.2 are equivalent.*

Proof. Consider the conditional density for position

$$f_l(z) \equiv \begin{cases} f_X(z) \text{ if } l = 0 \\ f_Y(z) \text{ if } l = 1 \end{cases} .$$

Using Bayes' formula for the joint densities yields

$$\begin{aligned} & f_{\theta(z)}(z,l) = f_{\theta_0}(z,l) \text{ a.e. } \forall l \in \{0,1\} \\ & \Leftrightarrow f_l(z)\mathbb{P}_{\theta_0}(l) = f(z)\mathbb{P}_{\theta_0}(l) \text{ a.e. } \forall l \in \{0,1\} \\ & \Leftrightarrow f_X = f_Y \text{ a.e..} \end{aligned}$$

□

3.2.4 Robust sequential p-value

Using the statement of Problem 3.2, we phrase our two-sample problem as a model selection problem and use a likelihood ratio test to obtain a p-value. The models we consider are distributions on length-n sequences, which can be obtained from sequential probability predictors. This approach has the advantage of providing a hypothesis test in which the sample size need not be fixed in advance as classical Neyman-Pearson does (see, e.g., [vdPG14, SSVV11]). More formally,

Theorem. 3.1. *Given some arbitrary distribution Q on infinite sequences $l_1, l_2, \ldots$ ($Q(l^n)$ denoting its marginal distribution on the first n outcomes),*[1] *a test that rejects $\mathtt{H}_0$ at any index n when the likelihood ratio*

$$\frac{\mathbb{P}_{\theta_0}(l^n)}{Q(l^n)} \leq \alpha \tag{3.6}$$

has a Type I error probability less or equal than α for problem 3.2, i.e.,

$$\mathbb{P}_{\theta_0}\left(\exists n : \frac{\mathbb{P}_{\theta_0}(L^n)}{Q(L^n)} \leq \alpha\right) \leq \alpha. \tag{3.7}$$

The likelihood ratio $\frac{\mathbb{P}_{\theta_0}(l^n)}{Q(l^n)}$ is called a robust *p-value.*

Proof. We consider the i.i.d. sequence L^n, Z^n and the class of models $f_{\theta(z)}(\cdot,\cdot)$ to which $f_{\theta_0}(\cdot,\cdot)$ belongs. Let us define $p_1(l^n, z^n) \equiv Q(l^n) f(z^n)$, which is a function from $\mathbb{R}^{d\cdot n} \times \mathcal{A}^n$ to $\mathbb{R}_0^+$. Note that

$$\frac{p_1(l^n, z^n)}{f_{\theta_0}(l^n, z^n)} = \frac{Q(l^n) f(z^n)}{\mathbb{P}_{\theta_0}(l^n) f(z^n)} = \frac{Q(l^n)}{\mathbb{P}_{\theta_0}(l^n)}.$$

Then, one has

$$\mathbb{E}_{\theta_0}\left[\frac{p_1(L^n, Z^n)}{f_{\theta_0}(L^n, Z^n)}\right] = \mathbb{E}_{\theta_0}\left[\frac{Q(L^n)}{\mathbb{P}_{\theta_0}(L^n)}\right] = \sum_{L^n \in \mathcal{A}^n} \mathbb{P}_{\theta_0}(L^n) \frac{Q(L^n)}{\mathbb{P}_{\theta_0}(L^n)} = 1$$

where the last inequality stems from Q being a distribution. Equation 3.7 follows from Theorem 3.5(Appendix 3.9). □

The following lemma shows that any sequential predictor assigning a probability to l_i using past label data l^{i-1}, and possibly all the available position data z^∞, complies with Thm. 3.1:

[1] When we write that Q is a distribution over infinite sequences $l_1, l_2, \ldots$, we mean that Q is a probability distribution relative to some sample space Ω and associated σ-algebra $\mathcal{F}$ such that $L_1, L_2, \ldots$ are all random variables on the probability triple $(\Omega, \mathcal{F}, Q)$ — see [Wil91, Section 4.5—4.8]. This automatically insures that the compatibility condition $\sum_{l' \in \mathcal{A}} Q(l^n, l') = Q(l^n)$ holds.

Lemma. 3.2. *Given a sequential probability predictor $Q\left(l_i|l^{i-1}\right)$ (i.e. $\sum_{l\in\mathcal{A}} Q\left(l|\cdot\right)=1$), one can build the following function of sequences $l^n\in\mathcal{A}^n$*

$$Q\left(l^n\right)\equiv\prod_{i=1}^{n}Q\left(l_i|l^{i-1}\right), \tag{3.8}$$

which is a distribution.

Proof. Let us prove by induction that it is a distribution using the distributive law to obtain the following expression

$$\sum_{l^n\in\mathcal{A}^n}Q\left(l^n\right)\equiv\sum_{l_1\in\mathcal{A}}Q\left(l_1|l^0\right)\sum_{l_2\in\mathcal{A}}Q\left(l_2|l^1\right)\cdots\sum_{l_n\in\mathcal{A}}Q\left(l_n|l^{n-1}\right)$$

where l^0 is the empty sequence. The base case corresponds to right-most sum and, since Q is a sequential probability predictor, we have

$$\sum_{l_n\in\mathcal{A}}Q\left(l_n|l^{n-1}\right)=1.$$

Observing that each sum is a convex combination of an expression that is equal to one proves the claim. □

3.2.5 Consistency via λ-pointwise universal distributions (λ-PUD)

We now define the requirements imposed on the distributions to obtain consistent tests, in a weaker sense that we call λ-consistency. To this end, we consider distributions depending on the position sequence z^n, whence the notation $Q\left(l^n|z^n\right)$.

Definition. 3.1. *Given $0<\lambda\leq 1$, a distribution Q is* λ-pointwise universal *(λ-PUD) if*

$$-\lim\frac{1}{n}\log Q\left(L^n|Z^n\right)\leq H\left(L|Z\right)-\log\lambda\ \ a.s.$$

for any $f_{\theta(z)}(\cdot,\cdot)$ generating the samples.

Before introducing the notion of λ-consistency, we prove the following lemma.

Lemma. 3.3. *Under the alternative hypothesis, $I\left(Z;L\right)>0$.*

Proof. By definition,

$$I\left(Z;L\right)\equiv-\int_{z\in\mathbb{R}^d}\sum_{l\in\mathcal{A}}f_{\theta(z)}(z,l)\log\frac{f_{\theta_0}(z,l)}{f_{\theta(z)}(z,l)}.$$

Recall that for a strictly convex function h and a random variable X, Jensen's lemma states that (see, e.g., [CT06, Thm. 2.6.2])

$$\mathbb{E}\left[h(X)\right]\geq h(\mathbb{E}\left[X\right]) \tag{3.9}$$

and the equality in 3.9 implies $X=\mathbb{E}\left[X\right]$ with probability 1.

We apply it to $X=\frac{f_{\theta_0}(Z,L)}{f_{\theta(z)}(Z,L)}$ and to the strictly convex function $h(X)=-\log(X)$. It remains to show that the random variable $\mathcal{X}$ is non-constant with non-null probability.

Under $\mathtt{H}_1$, there exists a set S of non-null probability such that $\forall z\in S\ \forall l\in\mathcal{A}$:

$$f_{\theta_0}(z,l)\neq f_{\theta(z)}(z,l)\Leftrightarrow\mathbb{P}_{\theta_0}\left(l\right)f(z)\neq\mathbb{P}_{\theta(z)}\left(l|z\right)f(z) \tag{3.10}$$

$$\Rightarrow\mathbb{P}_{\theta_0}\left(l\right)\neq\mathbb{P}_{\theta(z)}\left(l|z\right) \tag{3.11}$$

If the ratio were constant we should have

$$\frac{f_{\theta_0}(z,0)}{f_{\theta(z)}(z,0)}=\frac{f_{\theta_0}(z,1)}{f_{\theta(z)}(z,1)}\Leftrightarrow\frac{\mathbb{P}_{\theta_0}\left(0\right)}{\mathbb{P}_{\theta(z)}\left(0|z\right)}=\frac{\mathbb{P}_{\theta_0}\left(1\right)}{\mathbb{P}_{\theta(z)}\left(1|z\right)}=\frac{1-\mathbb{P}_{\theta_0}\left(0\right)}{1-\mathbb{P}_{\theta(z)}\left(0|z\right)}\Leftrightarrow\mathbb{P}_{\theta_0}\left(0\right)=\mathbb{P}_{\theta(z)}\left(0|z\right) \tag{3.12}$$

which contradicts Ineq. (3.11).

□

The following theorem introduces the λ-consistency property – from which one recovers the usual notion of consistency for $\lambda = 1$, and shows that this property is obtained from λ-PUD.

Theorem. 3.2 (λ-consistency)**.** *By Lemma 3.3, under the alternative hypothesis, $I(Z;L) = \varepsilon > 0$. Consider a λ-PUD Q such that $\lambda > 2^{-\varepsilon}$. Then, the test described in theorem 3.1 using that λ-PUD is consistent.*

Proof. The probability of rejecting $\mathtt{H}_0$ is

$$\mathbb{P}_{Z,L}\left(\exists n : \frac{\mathbb{P}_{\theta_0}(L^n)}{Q(L^n|Z^n)} \leq \alpha\right) \tag{3.13}$$

$$= \mathbb{P}_{Z,L}\left(\exists n : -\frac{1}{n}\log \mathbb{P}_{\theta_0}(L^n) + \frac{1}{n}\log Q(L^n|Z^n) \geq -\frac{\log \alpha}{n}\right). \tag{3.14}$$

Since Q is λ-PUD and, by the Asymptotic Equipartition Property (see, e.g., [CT06]), we have

$$\lim -\frac{1}{n}\log \mathbb{P}_{\theta_0}(L^n) + \frac{1}{n}\log Q(L^n|Z^n) \geq H(L) - H(L|Z) + \log\lambda \text{ a.s.}$$
$$= I(Z;L) + \log\lambda > 0$$

where the last inequality stems from $I(Z;L) > -\log\lambda$, which is a direct consequence of the assumptions.

Therefore, there exist $\delta > 0$ and $n'(\delta)$ and such that $\forall n \geq n'(\delta)$, the left-hand side of the inequality in Eq. (3.14) is greater than δ almost surely. For the right-hand side of the inequality, there exists $n''(\delta)$ such that $\forall n \geq n''(\delta)$, $-\frac{\log\alpha}{n} < \delta$.

Thus, for any $n \geq \max(n', n'')$, the inequality holds, and thus the probability of rejecting is 1. □

3.3 λ-Pointwise universal distributions via strongly pointwise consistent regressors

In this section, we construct λ-PUD thus yielding λ-consistent tests. The construction uses sequential predictors based upon strongly pointwise consistent regressors. These sequential predictors define a distribution via lemma 3.2, and therefore a robust sequential p-value, and yield a λ-PUD.

Sequential probability estimation from nonparametric regression. We first build a sequential predictor using nonparametric regression (see, e.g., [GK02]). Given a random vector (Z, R), where $Z \in \mathbb{R}^d$ and the response variable $R \in \mathbb{R}$, the *regression function* is defined as

$$m(z) = \mathbb{E}[R|Z = z]. \tag{3.15}$$

To obtain a sequential predictor, we consider the mapping $R = L$. This yields

$$m(z) = \mathbb{P}(L = 1|Z = z). \tag{3.16}$$

Let $m_n(z)$ be an estimate of $m(z)$ based on n i.i.d. realizations of (Z, R). Given some sequence of regression function estimators $\{m_n\}$ such that $0 \leq m_n(z) \leq 1\ \forall n, z$, let us define the following sequential predictor:

$$\hat{P}_i\left(l_i|l^{i-1}, z^i\right) \equiv m_{i-1}(z_i)\mathbb{1}_{l_i=1} + (1 - m_{i-1}(z_i))\mathbb{1}_{l_i=0}. \tag{3.17}$$

Notice that in this case, $\hat{P}_i$ predicts l_i based on l^{i-1} and z^i and ignores future z samples, and thus is compliant with lemma 3.2.

We are interested in the following sense of consistency since it allows to build λ-PUD.

Definition. 3.2. *A sequence of regression function estimates $\{m_n\}$ is* strongly pointwise consistent (s.p.c.) *if*

$$m_n(z) \xrightarrow{n\to\infty} m(z) \text{ a.s.} \tag{3.18}$$

for μ-almost all $z \in \mathbb{R}^d$, μ denoting the distribution of Z.

We call an s.p.c. sequence of regression estimates an s.p.c *regressor*. In [GK02, Sec. 25.6], some s.p.c. regressors are presented. For example, regressors based on partitioning, kernel and nearest neighbors are s.p.c under certain conditions for their parameters, when the absolute value $|R| < M$, for some M. Note that is our case since $R \in \{0, 1\}$. In our experiments, we use nearest neighbors regressors, which are described in Appendix 3.8.

Type I error and robust sequential p-value. We follow the construction of [Alg92][Sec. 2] in order to build a λ-PUD. For any distribution $Q(l|z)$ and $0 \leq \lambda < 1$, let us define

$$Q^{\lambda}(l|z) \equiv (1-\lambda)\frac{1}{|\mathcal{A}|} + \lambda Q(l|z). \tag{3.19}$$

Plugging $\hat{P}_n$ into the previous equation yields $\hat{P}_n^{\lambda}$, which, by lemma 3.2, can be turned into a distribution on length-n sequences and, thus, we have the following corollary.

Corollary. 3.1. *The test obtained with $\hat{P}_n^{\lambda}$ yields a robust p-value.*

Type II error and λ-consistency. Note that the constant term in Eq. (3.19) guarantees that the logarithm of Q^{λ} is bounded. This allows applying Breiman's Extended Ergodic Theorem [Bre57] to obtain the following theorem.

Theorem. 3.3. *$\hat{P}_n^{\lambda}$ based on a s.p.c. regressor yields a λ-PUD, thus a λ-consistent test. That is,*

$$\lim_n \frac{1}{n} \log \hat{P}_n^{\lambda}(L^n|Z^n) = \mathbb{E}\left[-\log \mathbb{P}_{\theta(z)}^{\lambda}(L|Z)\right] \quad a.s. \tag{3.20}$$

$$\leq H(L|Z) - \log\lambda \tag{3.21}$$

Proof. Let us define the doubly infinite i.i.d. process $(L,Z)_{-\infty}^{\infty} \equiv \ldots, (L_{-1}, Z_{-1}), (L_0, Z_0), (L_1, Z_1), \ldots$.

The claim

$$\lim_n \frac{1}{n} \log \hat{P}_n^{\lambda}(L^n|Z^n) = \mathbb{E}\left[-\log \mathbb{P}_{\theta(z)}^{\lambda}(L|Z)\right] \text{ a.s.} \tag{3.22}$$

is equivalent, by Lemma 3.2, to

$$\lim_n \frac{1}{n} \sum_{i=1}^{n} \log \hat{P}_i^{\lambda}\left(L_i|Z^i, L^{i-1}\right) = \mathbb{E}\left[-\log \mathbb{P}_{\theta(z)}^{\lambda}(L|Z)\right] \text{ a.s..} \tag{3.23}$$

Let us consider the operator T^i that shifts any sequence $\{\ldots, s_{-1}, s_0, s_1, \ldots\}$ by i positions to the left and let us denote

$$\hat{g}_i^{\lambda}((L,Z)_{-\infty}^{\infty}) \equiv -\log \hat{P}_{-i}^{\lambda}\left(L_1|Z_1, (L,Z)_{-i+1}^{0}\right) \tag{3.24}$$

where

$$\hat{P}_{-i}\left(L_1|Z_1, (L,Z)_{-i+1}^{0}\right) \equiv m_{-i}(Z_1)\mathbb{1}_{l_i=1} + (1-m_{-i}(Z_1))\mathbb{1}_{l_i=0} \tag{3.25}$$

where $m_{-i}(z)$ is an estimate of $m(z)$ based on $(L,Z)_{-i+1}^{0}$. Then the claim is equivalent to

$$\lim_n \frac{1}{n} \sum_{i=0}^{n-1} \hat{g}_i^{\lambda}(T^i(L,Z)_{-\infty}^{\infty}) = \mathbb{E}\left[-\log \mathbb{P}_{\theta(z)}^{\lambda}(L|Z)\right] \text{ a.s..} \tag{3.26}$$

Since $\hat{P}_n^{\lambda}$ is based on an s.p.c. regressor, we have that

$$\hat{P}_{-i}\left(l|z, (L,Z)_{-i+1}^{0}\right) \xrightarrow{i\to\infty} \mathbb{P}_{\theta(z)}(l|z) \text{ a.s.} \tag{3.27}$$

for all $l \in \mathcal{A}$ and f-almost all $z \in \mathbb{R}^d$. Since the measure of z values failing the previous convergence is null, one has

$$\mathbb{P}\left(\hat{P}_{-i}\left(L_1|Z_1, (L,Z)_{-i+1}^{0}\right) \xrightarrow{i\to\infty} \mathbb{P}_{\theta(z)}(L_1|Z_1)\right) = 1 \tag{3.28}$$

and thus

$$\mathbb{P}\left(\hat{P}^{\lambda}_{-i}\left(L_1|Z_1,(L,Z)^{0}_{-i+1}\right)\xrightarrow{i\to\infty}\mathbb{P}^{\lambda}_{\theta(z)}\left(L_1|Z_1\right)\right)=1. \tag{3.29}$$

Therefore, if we define

$$g^{\lambda}((L,Z)^{\infty}_{-\infty})\equiv-\log\mathbb{P}^{\lambda}_{\theta(z)}\left(L_1|Z_1\right), \tag{3.30}$$

we have that

$$\hat{g}^{\lambda}_{i}((L,Z)^{\infty}_{-\infty})\xrightarrow{i\to\infty}g^{\lambda}((L,Z)^{\infty}_{-\infty})\text{ a.s.}. \tag{3.31}$$

We also have that $\hat{g}^{\lambda}_{i}$ is bounded between 0 and $\log\frac{2}{1-\lambda}$, so that the claim follows from Breiman's extended ergodic theorem [Bre57].

The inequality 3.21 stems from $\mathbb{P}^{\lambda}_{\theta(z)}\left(L|Z\right)\geq\lambda\mathbb{P}_{\theta(z)}\left(L|Z\right)$.

□

As shown in [Alg92, Thm. 2], the following mixture is a pointwise universal distribution

$$\hat{P}^{\infty}_{n}\left(l^n|z^n\right)=\sum_{k=0}^{\infty}\mu_k\hat{P}^{\lambda_k}_{n}\left(l^n|z^n\right) \tag{3.32}$$

where $0\leq\lambda_k\nearrow 1$ and $\sum_{k=0}^{\infty}\mu_k=1$. Indeed, for each λ used in the mixture, one gets a λ-PUD by Lemma 3.4. Consequently, the mixture obtained for λ arbitrarily close to 1, yields a λ-PUD, approaching therefore the ideal case of $\lambda=1$. Yet, this mixture requires an infinite sum, which motivates our use of λ-PUD – which yield λ-consistent tests.

3.4 Increasing power using mixtures and switch distributions

Let us first consider Bayesian Model Averaging (BMA). Given a set of distributions $\{Q_k\}_k$ and weights μ_k such that $\sum_k\mu_k=1$, BMA produces the following *mixture*:

$$\mathrm{BMA}_{\{Q_k\}}\left(l^n|z^n\right)\equiv\sum_{k}\mu_kQ_k\left(l^n|z^n\right). \tag{3.33}$$

Pointwise universal mixtures. We first prove a lemma that allows building λ-consistent two-sample tests by mixing one λ-PUD with arbitrary distributions:

Lemma. 3.4. *A mixture of distributions containing at least one λ-PUD Q_0 with weight $\mu_0(n)$ such that $\log\mu_0(n)=o(n)$ yields a λ-PUD.*

Proof. Consider a λ-PUD Q_0 and the arbitrary distributions $Q_k\left(l^n|z^n\right),k>0$. Assuming that $\sum_k\mu_k=1$, we define the mixture:

$$Q\left(l^n|z^n\right)\equiv\sum_{k}\mu_kQ_k\left(l^n|z^n\right).$$

Since $Q\left(l^n|z^n\right)\geq\mu_0(n)Q_0\left(l^n|z^n\right)$, one has

$$\begin{aligned}-\lim\frac{1}{n}\log Q\left(L^n|Z^n\right)&\leq-\lim\frac{1}{n}\log\mu_0(n)Q_0\left(L^n|Z^n\right)\\&=-\lim\frac{1}{n}\log Q_0\left(L^n|Z^n\right)+\frac{\log\mu_0(n)}{n}\qquad\leq H\left(L|Z\right)-\log\lambda\text{ a.s.}.\end{aligned}$$

□

An interesting application of mixtures is the following one. Since it is not clear which neighborhood size function k_n is best for a KNN regressor, we consider a set of functions $k_n=n^p$, where p takes values in some set – finite for practical purposes. All these predictors yield λ-PUD if $p<1$ (see Appendix 3.8). But using a mixture allows one to favor the best size function by updating the a posteriori weights according to past performance.

Remark 3.3. *Compare this to the more rigid approach of splitting the sample in two and optimizing on one part and using the optimized value to test on the other (as for TreeRank [CDV09] or optimized kernel MMD [GSS+12, ZGB13]).*

Note also that our process keeps learning beyond the index of the sample corresponding to the training set size.

Remark 3.4. *Another interest of mixtures is to use non universal distributions that could be tailored to some more specific situations on which they may exhibit a better power than the universal ones. For example, one could use a linear kernel support vector machine*[2] *which would be expected to be more sensitive on distributions that differ by a shift.*

If a better finite size performance is sought, other predictors could be plugged in our framework, which could be better for different kind of differences between distributions.

Model switching. The mixture of lemma 3.4 also allows model switching to avoid the catch-up phenomenon (see [vEGdR12]). That is, even if we are under $\mathtt{H}_1$, when few samples are available it can be better to predict using $\mathbb{P}_{\theta_0}$ and then switch to $\hat{P}_n$ when more samples are available.

Let us define the following distribution that switches before time s from $\mathbb{P}_{\theta_0}$ to any predictor $\hat{P}_n$:

$$\hat{P}^{Sw}_{\{\mathbb{P}_{\theta_0},\hat{P}_n\}}\left(l^n|s,z^n\right) \equiv \mathbb{P}_{\theta_0}\left(l^{s-1}\right)\hat{P}_n\left(l^n_s|l^{s-1},z^n\right) \tag{3.34}$$

where l^0 is the empty sequence and $\mathbb{P}_{\theta_0}\left(l^0\right) \equiv 1$. Given a prior $\pi(s)$ on the switching time s, we define the following switch distribution

$$\hat{P}^{Sw(\pi)}_{\{\mathbb{P}_{\theta_0},\hat{P}_n\}}\left(l^n|z^n\right) \equiv \sum_s \pi(s)\hat{P}^{Sw}_{\{\mathbb{P}_{\theta_0},\hat{P}_n\}}\left(l^n|s,z^n\right). \tag{3.35}$$

The following lemma follows from lemma 3.4.

Lemma. 3.5. *Given a λ-PUD $\hat{P}_n$, the switch distribution defined by Eq. (3.35) with a prior π such that $\log\pi(0) = o(n)$ is λ-PUD.*

A first option for the switch time prior is the horizon-free prior defined in [vEGdR12]

$$\pi_S(s) \equiv \frac{1}{s(s+1)}. \tag{3.36}$$

Another option, when the horizon n is known, is the uniform prior

$$\pi_U(s) \equiv \frac{1}{n}. \tag{3.37}$$

Note that Eq. (3.34) can be interpreted as a training phase (until s) followed by a test phase that keeps learning. Eq. (3.35) replaces the choice of s by a mixture of all possible values, weighted by a prior.

3.5 Implementation and complexity

Here, let n be the number of samples effectively processed until the process is stopped.

KNN The KNN predictor was implemented using exact brute force FLANN's implementation [ML09] whose time complexity is linear in the number of samples seen so far. When a new sample z_i is seen, its nearest neighbors are searched to compute the labels' probabilities as described in Appendix 3.8, then the pair (l_i, z_i) is added to the training set. At the beginning, when the training set is empty, it predicts using $\mathbb{P}_{\theta_0}$. Therefore, the time complexity is quadratic in n and the space complexity is linear in n.

Since the algorithms we describe next are independent on whether the predictors depend on z^n or not, we omit z^n from notation.

[2] using, e.g., Platt's scaling to obtain probabilities.

BMA The sequential formula for BMA is, by Bayes rule,

$$\mathrm{BMA}_{\{Q_k\}}\left(l_i|l^{i-1}\right) = \sum_k Q_k\left(l_i|l^{i-1}\right)\mu(k|l^{i-1}) \tag{3.38}$$

where the weights correspond to the predictors' posterior probability, i.e.,

$$\mu(k|l^{i-1}) = \frac{\mu_k Q_k\left(l^{i-1}\right)}{\sum_k \mu_k Q_k\left(l^{i-1}\right)}. \tag{3.39}$$

Therefore, it suffices to sequentially maintain the total probability of each predictor using a log representation for probabilities to avoid numerical problems. More precisely, the following identities allow to multiply and to add two given probabilities x and y in log representation:

$$\begin{aligned}\log(xy) &= \log(x) + \log(y)\\ \log(x+y) &= \log(x) + \log(1+\exp(\log(y)-\log(x))).\end{aligned}$$

Practically, for the sum, if the difference in magnitude between x and y is too large[3], the result is simply the larger number.

Therefore, the BMA algorithm is linear in time and uses constant space.

Switching In order to perform sequential computation of the switch distribution we use the algorithm described in [E+10] which is a special case of the Forward algorithm of [Rab89]. In a nutshell, it is like BMA except that weights are updated in a different manner. Algorithm 1 describes it for our case. It takes as argument the prior π in the conditional form $\mu(s) \equiv \pi(T = s|T \geq s)$, T representing the switching time, which verifies

$$\begin{aligned}\pi(s) = \pi(T=s) &= \pi(T=s|T\geq s)\pi(T\geq s)\\ &= \pi(T=s|T\geq s)\prod_{i=1}^{s-1}(1-\pi(T=i|T\geq i)).\end{aligned}$$

The standard and uniform priors presented in Section 3.4 yield respectively

$$\begin{cases}\mu_S(s) &= \frac{1}{s+1}\\ \mu_U(s) &= \frac{1}{n-s+1}\end{cases}, s \geq 1.$$

As BMA, the switching algorithm is linear in time and uses constant space.

3.6 Experiments

3.6.1 Instantiations

Our constructions allow defining two-sample tests from λ-PUD in general (section 3.2), and s.p.c. regressors in particular (section 3.3). More precisely, we define the following λ-consistent two-sample tests:

1. $\mathrm{KNN}_p \equiv \hat{P}_n^\lambda$ obtained via KNN with $k_n = \lceil n^p \rceil$ and $\lambda = 0.9999$
2. $\mathrm{BMA} \equiv \mathrm{BMA}_{\{\mathrm{KNN}_p\}}, p \in \{.3, .5, .7, .9\}$ with uniform prior μ
3. $\mathrm{SW}_{\pi_U} \equiv \hat{P}^{Sw(\pi_U)}_{\{\mathbb{P}_{\theta_0},\mathrm{BMA}\}}$
4. $\mathrm{SW}_{\pi_S} \equiv \hat{P}^{Sw(\pi_S)}_{\{\mathbb{P}_{\theta_0},\mathrm{BMA}\}}$

For tests **3.** and **4.**, we consider two versions: the first one computes the likelihood ratio on the full sequence generated by the random device discussed in section 3.2.2 (whence the letter F for Full); the second one exploits the $\exists$ quantifier of Eq. (3.7), i.e., stops at the first index such that the likelihood ratio is below α (whence the letters OS for Optional Stop). Note that for tests **1.** and **2.**, we only report results for the full version.

[3] In our experiments, we check whether $|\log(y) - \log(x)| > 100$.

Algorithm 1 Sequential Switch Distribution Computation

{**Input:** a sequence l^n, a conditional prior $\mu(s) \equiv \pi(T = s | T \geq s)$ }
{**Output:** $\hat{P}^{Sw(\pi)}_{\{\mathbb{P}_{\theta_0}, \hat{P}_n\}}(l^n)$ }

$p \leftarrow 1$
{**Initialize weights:**}
{no switch before time 1}
$w_{\theta_0} \leftarrow 1 - \mu(1)$
{switch before time 1}
$w_{\theta(z)} \leftarrow \mu(1)$
for $i = 1 \dots n$ **do**
  {**Predict:**}
  $p \leftarrow p \cdot \left(w_{\theta_0} \mathbb{P}_{\theta_0}\left(l_i | l^{i-1}\right) + w_{\theta(z)} \hat{P}_{i-1}\left(l_i | l^{i-1}\right) \right)$
  {**Update posterior weights:**}
  {no switch before time $i+1$}
  $w_{\theta_0} \leftarrow w_{\theta_0} \mathbb{P}_{\theta_0}\left(l_i | l^{i-1}\right)(1 - \mu(i+1))$
  {switch at time i or already switched before }
  $w_{\theta(z)} \leftarrow w_{\theta_0} \mathbb{P}_{\theta_0}\left(l_i | l^{i-1}\right)\mu(i+1) + w_{\theta(z)} \hat{P}_{i-1}\left(l_i | l^{i-1}\right)$
  {normalize weights}
  $w_{\theta_0} \leftarrow {}^{w_{\theta_0}}/_{(w_{\theta_0} + w_{\theta(z)})}$
  $w_{\theta(z)} \leftarrow {}^{w_{\theta(z)}}/_{(w_{\theta_0} + w_{\theta(z)})}$
return p

3.6.2 Contenders

We compare their performance against the following methods:

- MMD_b: the Maximum Mean Discrepancy two-sample test presented in [GBR+12] using the bootstrap approach to compute the rejection region. We used the MATLAB code available at `http://www.gatsby.ucl.ac.uk/~gretton/mmd/mmd.htm`. 500 shuffles were used for the bootstrap.

- TRank: the AUC optimization based two-sample test presented in [CDV09]. We used the R implementation available from CRAN Archive at `http://cran.r-project.org/src/contrib/Archive/TreeRank` with the default parameters. Since we test our methods on the same distributions and sample sizes as in [CDV09] we also include the performance stated in that paper, which we denote as $\text{TRank}_{\text{ref}}$. In those cases, we ran TRank with same split value as $\text{TRank}_{\text{ref}}$.

- Kernel optimized MMD (OPT, MxRt, MxMMD, L2, xvalc and med): these are the methods proposed in [GSS+12]. They are all based on splitting the data in train/test sets, performing some kernel optimization on the train set then testing on the test set. The kernels considered are Gaussian ones with width $\sigma \in \{2^i\}_{i=-15\dots10}$.

3.6.3 Results

Setup. We set $\alpha = 0.05$. Experiments were run with $n_0 = n_1$ since the implementations of selected contenders (MMD_b) require this balance. Therefore, we set $\theta_0 = 1/2$ in order to match these proportions (see discussion in section 3.2.2).

Following [CDV09, GSS+12], our experiments assess the type I and type II errors of our tests and their contenders, under various conditions: Gaussian data in low and medium dimension, mixture of Gaussian data aiming at assessing the incidence of *scale*.

Gaussian data in dimension $d = 4$. Table 3.1 presents the results on data generated using the specification of [CDV09, Fig. 1], corresponding to different 4-dimensional Gaussians:

- Ex1 : two populations drawn from the same distribution, so that Type I error is assessed. Sample sizes: $n_0 = n_1 = 1000$.

- Ex2 : two populations drawn from two shifted Gaussians, so that Type II error is assessed. Sample sizes: $n_0 = n_1 = 1000$.

- Ex3 : corresponds to two subtly shifted Gaussians. Sample sizes: Ex3a: $n_0 = n_1 = 3000$; Ex3b: $n_0 = n_1 = 5000$.

Regarding Ex1 (Table 3.1, line 1), our tests using the full sequence are conservative – no type I error, an observation also valid in the other experiments. For the remaining experiments (Table 3.1, lines 2-4), we observe that the switching distribution with a uniform prior is the best amongst our predictors. Also, its power is comparable to that of TRank, yet worse than that of MMD_b (Ex3b).

Gaussian data in dimension $d = 10, 30$. Table 3.2 presents the results on data generated using the same specifications of [CDV09][Fig. 2], which correspond to 10 and 30-dimensional gaussians shifted by $\Delta\mu = 0.2$ with $n_0 = n_1 = 2000$. Here we estimate Type II error probability.

Although individual predictors KNN_i yield the worst performance, SW_{π_U} shows good performances, even outperforming TRank in dimension 10 – in our replica, which use default parameters. We notice, though, that performances degrade upon increasing the dimension, a likely consequence of distance concentration phenomena perturbing the choice of neighbors by KNN.

Lattice data and incidence of the scale. Table 3.3 presents the results on data corresponding to the specification of [GSS+12]: two 5×5 grids of two-dimensional Gaussians (a.k.a. blobs) that differ in covariance (parameters: stretch=10, rotation_angle=$\pi/4$, blob_distance=15). We consider three cases:
– **B1:** two populations drawn from the mixture of both blobs, so that Type I error is assessed. Sample sizes: $n_0 = n_1 = 200$.
– **B2:** two populations drawn from each of the blobs, so that Type II error is assessed. Sample sizes: $n_0 = n_1 = 1500$.
– **B3:** Same as B2 but with larger sample sizes: $n_0 = n_1 = 2000$.

As noted in [GSS+12], it is a prototypical example where MMD_b fails and kernel width optimization is important, since differences occur at a smaller scale. For our predictors, this scale is captured by slower k_n. It is important to emphasize, that predictors with larger k_n are also consistent and, therefore, they would also detect the differences with more samples. Remarkably, when $n_0 = n_1 \geq 1500$, SW_{π_U} (F or OS) outperforms all the MMD contenders.

3.7 Conclusion

This work introduces the first sequential nonparametric two-sample test, based on sequential prediction of labels defining the two populations. Our test is shown to be consistent when prediction is carried out by strongly pointwise consistent regressors. We show that mixtures of distributions based on KNN regressors are effective in favoring the best neighborhood size function. This update being carried out along the sequential process is more flexible than classical approaches splitting the data into a training and test sets. We also show that model switching increases the power, a fact related to the ability of automatically selecting the best splitting point in a train/test paradigm. Experimentally, while no test is expected to be the most powerful for all kinds of data, our best constructs outperform state-of-the-art ones on selected challenging datasets.

We foresee two outstanding questions. Complexity-wise, the regressors used rely on exact nearest neighbor queries, exhibiting linear complexity in the worst-case. Inferring whether our tests remain consistent when information of lower quality is used (e.g. approximate nearest neighbors) would allow using more efficient data structures. In addition, obtaining consistency guarantees under constant size memory requirements would be of special interest in a streaming context.

Table 3.1 Gaussian data in dimension $d = 4$. The symbol $\mathtt{H}_0$ or $\mathtt{H}_1$ at the beginning of each line indicates whether the null is true or false. Numbers indicate the percentage of trials for which the null hypothesis was rejected, given $\alpha = 0.05$. A total of 150 trials were done.

Case	KNN$_{.3}$	KNN$_{.5}$	KNN$_{.7}$	KNN$_{.9}$	BMA	SW$^{(F)}_{\pi_U}$	SW$^{(F)}_{\pi_S}$	SW$^{(OS)}_{\pi_U}$	SW$^{(OS)}_{\pi_S}$	TRank	TRank$_{\text{ref}}$	MMD$_b$
$\mathtt{H}_0$, Ex1	0	0	0	0	0	0	0	2	6	4.7	1	3.3
$\mathtt{H}_1$, Ex2	5.3	100	100	100	100	100	100	100	100	100	99	100
$\mathtt{H}_1$, Ex3a	0	0	0	0	0	18.7	5.3	20	9.3	40.7	45	90
$\mathtt{H}_1$, Ex3b	0	0	1.3	0	0.7	48.7	16	65.3	21.3	76	73	98.7

Table 3.2 Gaussian data in dimension $d = 10, 30$. For conventions, see the caption of Table 3.1. In this experiment, the data under $\mathtt{H}_0$ was generated by sampling from the mixture of both distributions. 100 trials were done. (NB: the conditions associated with TRank$_{\text{ref}}$ are specified in M. Depecker's PhD Thesis [Dep10].)

Case	KNN$_{.3}$	KNN$_{.5}$	KNN$_{.7}$	KNN$_{.9}$	BMA	SW$^{(F)}_{\pi_U}$	SW$^{(F)}_{\pi_S}$	SW$^{(OS)}_{\pi_U}$	SW$^{(OS)}_{\pi_S}$	TRank	TRank$_{\text{ref}}$	MMD$_b$
$\mathtt{H}_0, d = 10$	0	0	0	0	0	1	0	1	5	5		6
$\mathtt{H}_0, d = 30$	0	0	0	0	0	0	0	0	3	6		6
$\mathtt{H}_1, d = 10$	0	0	0	0	0	69	20	68	35	36	90	99
$\mathtt{H}_1, d = 30$	0	0	0	0	0	22	5	29.5	10	25	85	95

Table 3.3 Lattice of Gaussians in dimension $d = 2$. For conventions, see the caption of Table 3.1. The data under $\mathtt{H}_0$ was generated by sampling from the mixture of both distributions. A total of 200 trials were done.

Case	KNN$_{.3}$	KNN$_{.5}$	KNN$_{.7}$	KNN$_{.9}$	BMA	SW$^{(F)}_{\pi_U}$	SW$^{(F)}_{\pi_S}$	SW$^{(OS)}_{\pi_U}$	SW$^{(OS)}_{\pi_S}$
$\mathtt{H}_0$, B1	0	0	0	0	0	0	0	0.5	5.5
$\mathtt{H}_1$, B2	11	0.5	0	0	11	21.5	14	27.5	16.5
$\mathtt{H}_1$, B3	78.5	62.5	0	0	85	93	89.5	96.5	91.5

Case	TRank	MMD$_b$	OPT	MxRt	MxMMD	L2	xvalc	med
$\mathtt{H}_0$, B1	6	5	4	2.5	2.5	4	7	7
$\mathtt{H}_1$, B2	100	6	18.5	15.5	15	14.5	15.5	5
$\mathtt{H}_1$, B3	100	6.5	21	18.5	15	15	21	7.5

3.8 Appendix: Nonparametric regression based on k_n-nearest neighbors

Here we describe the s.p.c. k_n-nearest neighbor regression function estimate (see [GK02, Ch.6&25] for further details). Given the training data $\{Z_i, R_i\}_{i=1,\dots,n}$, let us denote as $R_{(i,n)}(z)$ the response value corresponding to i-th nearest neighbor (with some tie-breaking rule) of z in Z^n. Then, the k_n-nearest neighbor (k_n-NN) regression function estimate is defined by

$$m_n(z) = \frac{1}{k_n} \sum_{i=1}^{k_n} R_{(i,n)}(z). \tag{3.40}$$

Then we have the following theorem [GK02, Thm. 25.17]:

Theorem. 3.4 (Strong pointwise consistency of k-NN)**.** *If $|R| < C$ for some $C < \infty$,*

$$\frac{k_n}{\log n} \to \infty \text{ and } \frac{k_n}{n} \to 0,$$

then the k_n-NN estimate using Euclidean distance is strongly pointwise consistent.

3.9 Appendix: Optional Stopping

In order to prove Theorem 3.1, we use the following result

Theorem. 3.5 ([vdPG14, Thm. 3.1], special case of [SSVV11])**.** *Given a sample space $\mathcal{X}$ and a set of parameters M, suppose we have data $X_1, \dots, X_n$ i.i.d. $\sim \mathbb{P}_\mu, X_i \in \mathcal{X}, \mu \in M$, and a test rejects the null hypothesis $\mathtt{H}_0 : \mu = \mu_0$ if*

$$\frac{\mathbb{P}_{\mu_0}(x^n)}{p_1(x^n)} \leq \alpha, \tag{3.41}$$

where α is a constant and p_1 is a function from $\mathcal{X}^n$ to $\mathbb{R}_0^+$. If $p_1(x^n)$ is such that, for all n, the compatibility condition

$$\sum_{x' \in \mathcal{X}} p_1(x^n, x') = p_1(x^n) \tag{3.42}$$

holds and $\mathbb{E}_{\mu_0}\left[\frac{p_1(X^n)}{\mathbb{P}_{\mu_0}(X^n)}\right] \leq 1$, then:

$$\mathbb{P}_{\mu_0}\left(\exists n : \frac{\mathbb{P}_{\mu_0}(X^n)}{p_1(X^n)} \leq \alpha\right) \leq \alpha. \tag{3.43}$$

Chapter 4

A Sequential Non-parametric Two-Sample Test via Random Projection Context Trees

4.1 Introduction

In this chapter, we propose a sequential two-sample test using sequential conditional probability estimation based on spatial partitioning instead of nearest neighbors as in Chapter 3. That is, we base again our test on the result of theorem 3.1 but we propose a different construction for Q based on a discretization of Z.

A remarkable feature of this test is its semi-supervised nature, meaning that, in an initial step, it uses an unlabeled set for the spatial partitioning. Then, its uses a sequence of labeled samples as in Chapter 3. The unlabeled set can be, for example, a subset of the labeled one. We call our construction of Q *Random Projection Context Trees* since it uses the Random Projection Trees of [DF08] to define a spatial partitioning whose cells are used as context to condition the probability of the labels.

The construction is better described in three steps corresponding to the following sections. The first one consists in a discretization of the $\mathbb{R}^d$ using a spatial partition where in each cell a distribution is estimated. The second step consists in considering a hierarchical set of random partitions (obtained from the unlabeled set) each providing a discretized conditional distribution from the first step, and then mixing them using a weighting scheme. Practically, the hierarchical partition is based on a binary tree. The success of this construction depends on the ability of the random partitions to reveal cells with a proportion of label 0 different from the unconditional probability θ_0. In the third step, we boost the probability of finding such cells by considering and ensemble of trees.

Complexity-wise, processing a new labeled sample has logarithmic time complexity w.r.t. the unlabeled set size, and negligible memory footprint, which makes it suitable for streaming data. Moreover, this construction exhibits automatic adaption to local scale and does not require parameters to be set.

Finally, we present experiments stressing the power of our test and its ability to cope with different scales.

4.2 Discretizing the sample space

4.2.1 Spatial partition and discretized distribution

A spatial partition $P = \{C_1, C_2, \ldots, C_{|P|}\}$ of $\mathbb{R}^d$ is a finite collection of cells of mutually disjoint interior, and whose union equals $\mathbb{R}^d$. Averaging $\theta(z) \equiv \mathbb{P}(L = 0|Z = z)$ yields

$$\theta_{C_i} \equiv \mathbb{P}(L = 0|Z \in C_i) = \frac{\int_{z \in C_i} \theta(z) f_Z(z) dz}{\int_{z \in C_i} f_Z(z) dz}, \tag{4.1}$$

from which one defines the following vector characterizing the partition:

$$\boldsymbol{\theta}_P = [\theta_{C_1}, \ldots, \theta_{C_{|P|}}]. \tag{4.2}$$

Let $C_P(z)$ denote the cell of P in which a point z falls, then we denote

$$\mathbb{P}_{\boldsymbol{\theta}_P}(l|z) \equiv \mathbb{P}(L = l | Z \in C_P(z)), \tag{4.3}$$

and the associated joint distribution

$$\mathbb{P}_{\boldsymbol{\theta}_P}(l^n|z^n). \tag{4.4}$$

4.2.2 Consistency via pointwise universal distributions on a revealing partition

In order to be able to detect a change, the partition P must reveal differences between the unconditional and the conditional distributions. Therefore, we define the following.

Definition. 4.1. *A* revealing cell C *is a cell such that* $\theta_C \neq \theta_0$. *A* revealing partition P *is such that it contains at least one revealing cell.*

Lemma. 4.1. *Let* P *be a revealing partition, then*

$$H_{\boldsymbol{\theta}_P}(L|Z) \equiv -\sum_{C \in P} \mathbb{P}(Z \in C) \sum_{l \in \{0,1\}} \mathbb{P}_{\boldsymbol{\theta}_P}(l|z) \log \mathbb{P}_{\boldsymbol{\theta}_P}(l|z) < H(L) \tag{4.5}$$

where $H(L)$ *denotes the entropy of* L.

Proof. Since conditioning reduces entropy on average, we have that $H_{\boldsymbol{\theta}_P}(L|Z) \leq H(L)$ with equality if and only if variables L and $C_P(Z)$ are independent, i.e., $\forall l, c$, $\mathbb{P}(L = l | C_P(Z) = c) = \mathbb{P}(L = l)$ which is not the case since P is a revealing partition and thus there exists a cell c such that the equality does not hold. □

On the way to defining a distribution on infinite sequences (thus compatible with theorem 3.1) and also yielding a consistent test, we define:

Definition. 4.2. *A distribution* Q *is* pointwise universal *with respect to a partition* P *(denoted as* P*-PUD) if*

$$\lim_{n \to \infty} -\frac{1}{n} \log Q(L^n|Z^n) \leq H_{\boldsymbol{\theta}_P}(L|Z) \ \ a.s.. \tag{4.6}$$

The following theorem guarantees consistency when P-PUD are used:

Theorem. 4.1 (Consistency)**.** *The test described in Theorem 3.1 with* Q *being a distribution on infinite sequences and a* P*-PUD is consistent.*

Proof. The probability of rejecting $\mathtt{H}_0$ is

$$\mathbb{P}_{Z,L}\left(\exists n : \frac{\mathbb{P}_{\theta_0}(L^n)}{Q(L^n|Z^n)} \leq \alpha\right) \tag{4.7}$$

$$= \mathbb{P}_{Z,L}\left(\exists n : -\frac{1}{n} \log \mathbb{P}_{\theta_0}(L^n) + \frac{1}{n} \log Q(L^n|Z^n) \geq -\frac{\log \alpha}{n}\right). \tag{4.8}$$

Since Q is a P-PUD and, by the Asymptotic Equipartition Property (see, e.g., [CT06]), we have

$$\lim -\frac{1}{n} \log \mathbb{P}_{\theta_0}(L^n) + \frac{1}{n} \log Q(L^n|Z^n) = H(L) - H_{\boldsymbol{\theta}_P}(L|Z) \ \text{a.s..}$$

And by lemma 4.1, the last expression is strictly positive.

Therefore, there exist $\delta > 0$ and $n'(\delta)$ and such that $\forall n \geq n'(\delta)$, the left-hand size of the inequality in Eq. (4.8) is greater than δ almost surely. For the right-hand side of the inequality, there exists $n''(\delta)$ such that $\forall n \geq n''(\delta)$, $-\frac{\log \alpha}{n} < \delta$.

Thus, for any $n \geq \max(n', n'')$, the inequality holds, and thus the probability of rejecting is 1. □

4.2.3 A P-PUD based on Jeffreys mixture

Consider a cell C of a partition P. The number of samples in that cell and the number of samples with a specific label are denoted by

$$n(C) \equiv \#\{l_j : z_j \in C, j = 1, \ldots, n\} \tag{4.9}$$

$$n_l(C) \equiv \#\{l_j = l : z_j \in C, j = 1, \ldots, n\}. \tag{4.10}$$

Given a cell C, θ_C can be estimated by *Maximum Likelihood (ML)*, in which case it is equal to the fraction of samples within cell C having the label 0, i.e.,

$$\hat{\theta}_C \equiv \frac{n_0(C)}{n(C)}. \tag{4.11}$$

Nevertheless, ML does not produce a distribution over $\{0,1\}^n$ as required by Thm. 3.1. It can be corrected using Normalized Maximum Likelihood (NML) [Sht87, Ris96] but it is hard to compute. Instead, we shall use the following Bayesian mixture, which is asymptotically equivalent to NML (when taking the normalized logarithm) :

$$\hat{P}_n^{\text{Jeffreys}(C)}\left(l^n|z^n\right) \equiv \int_{\theta\in[0,1]} \theta^{n_0(C)}(1-\theta)^{n_1(C)} w(\theta) d\theta \tag{4.12}$$

with $w(\cdot)$ being Jeffreys' prior for the Bernoulli distribution [Jef46] defined, in this case, as

$$w(\theta) \equiv \text{Beta}\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{\pi\sqrt{\theta(1-\theta)}}. \tag{4.13}$$

Using the gamma function Γ, it can be expressed as follows

$$\hat{P}_n^{\text{Jeffreys}(C)}\left(l^n|z^n\right) = \frac{\Gamma\left(n_0(C)+\frac{1}{2}\right)\Gamma\left(n_1(C)+\frac{1}{2}\right)}{\Gamma(n(C)+1)\pi}. \tag{4.14}$$

Combining these distributions for all the cells of the partition P yields a distribution on the whole sequence of labels:

$$\hat{P}_n^{\text{Jeffreys}(P)}\left(l^n|z^n\right) \equiv \prod_{C\in P} \hat{P}_n^{\text{Jeffreys}(C)}\left(l^n|z^n\right). \tag{4.15}$$

On the way to establishing consistency, we prove the following

Lemma. 4.2 (Jeffreys Mixture is P-PUD)**.** *$\hat{P}_n^{\text{Jeffreys}(P)}$ is a P-PUD, i.e.,*

$$-\frac{1}{n}\log \hat{P}_n^{\text{Jeffreys}(P)}\left(L^n|Z^n\right) \leq -\frac{1}{n}\log \mathbb{P}_{\boldsymbol{\theta}_P}\left(L^n|Z^n\right) + |P| O\left(\frac{\log n}{2n}\right) \tag{4.16}$$

$$\xrightarrow{n\to\infty} H_{\boldsymbol{\theta}_P}\left(L|Z\right) \ \ a.s. \tag{4.17}$$

Proof. The uniform bound of [WST95, Eq.11], states that for any cell C of the partition and any $\theta \in [0,1]$

$$\log \frac{\theta^{n_0(C)}(1-\theta)^{n_1(C)}}{\hat{P}_n^{\text{Jeffreys}(C)}\left(l^n|z^n\right)} \leq \frac{1}{2}\log n(C) + 1. \tag{4.18}$$

Therefore, for any cell C of the partition P

$$-\frac{1}{n}\log \hat{P}_n^{\text{Jeffreys}(C)}\left(l^n|z^n\right) \leq -\frac{1}{n}\log \theta^{n_0(C)}(1-\theta)^{n_1(C)} + \frac{1}{2n}\log n(C) + \frac{1}{n}. \tag{4.19}$$

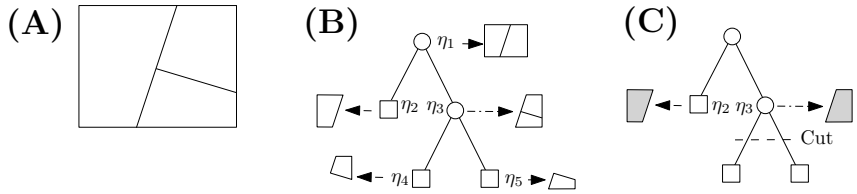
Then, since $\sum_{C\in P} n(C) = n$, using the log sum inequality, we have that

$$\sum_{C\in P} \log n(C) = -\sum_{C\in P} 1 \log \frac{1}{n(C)} \leq |P| \log \frac{n}{|P|}. \tag{4.20}$$

Then, by adding inequality 4.19 for each cell of the partition and taking, in each cell C_i, $\theta = \theta_{C_i}$ from $\boldsymbol{\theta}_P$, we obtain the inequality of the statement.

The limit follows from the Asymptotic Equipartition Property (see, e.g., [CT06]). □

Figure 4.1 Tree and partitions (A) Partition of a 2D domain by a binary space partition tree, with two levels. **(B)** The corresponding full tree. To each node is associated a region; internal nodes also come with a splitting rule to define the regions of the two children. Note that a tree T generates a set of partitions. Identifying a node with its associated region, these partitions are $\mathcal{P}_T = \{\eta_1, \{\eta_2, n_3\}, \{\eta_2, \eta_4, \eta_5\}\}$. **(C)** The partition associated with the dashed cut consists of the two grayed cells.



Corollary. 4.1. *Let P be a revealing partition. Then the test of Theorem 3.1 with the distribution $\hat{P}_n^{\text{Jeffreys}(P)}$ is consistent.*

Proof. Follows from P being a revealing partition, Lemma 4.2 and Theorem 4.1. □

Remark 4.1 (Sequential computation of Jeffreys' Mixture)**.** *Let $n_l^i(C(z))$ denote the number of times the symbol l occurred in the sequence l^i restricted to samples that fell in the cell containing z and let $n^i(C(z))$ denote the total number of symbols of the sequence l^i restricted to samples that fell in the cell containing z. Then, Jeffreys' mixture can be computed sequentially using Kritchevsy-Trofimov (KT) estimator [KT81], i.e,*

$$\hat{P}_n^{\text{Jeffreys}(P)}(l^n|z^n) = \prod_{i=1}^{n} \frac{n_{l_i}^{i-1}(C(z_i)) + \frac{1}{2}}{n^{i-1}(C(z_i)) + \frac{|\mathcal{A}|}{2}}. \tag{4.21}$$

Therefore, the sequential computation only requires maintaining the counters $n_l^i(C)$ for each cell C.

4.3 Using a hierarchical ensemble of partitions associated with a tree

4.3.1 Hierarchical partitioning based on random projection trees

Practically, we consider a class of partitions associated to full binary space partition trees.[1] In such a tree, each internal node has two children, be they internal nodes or leaves, obtained by recursively applying a splitting rule (Fig. 4.1(A,B)). To define such partitions, let a *tree cut* be a simplification yielding a full binary tree. A partition consists of the cells associated with the leaves of a tree cut (Fig. 4.1(C)). If we consider a perfect depth-K binary tree,[2] there exists a doubly-exponential number of $N_K \approx 1.5^{2^K}$ partitions embedded within (see, e.g., [KSB08]).

Random Projection (RP) Trees create a data adaptive hierarchical partition P of $\mathbb{R}^d$ by using projections and splittings along random lines [DF08]. We use the RPTree-Max version, which splits the data at the median plus a random jitter. This construction is suitable for high-dimensional data that has lower intrinsic dimension. More precisely, given a cell $C \in P$ of doubling dimension d' [Ass83], with probability at least $1/2$, only $O(d' \log d')$ levels are required to halve the diameter of the data contained in C.

In our implementation, the trees are based on a set of N unlabeled samples and we split until there is only one point per cell. Although the unlabeled set used for tree construction can be an arbitrary one, in the experiments, we use all the positions occurring in the sequence to be considered. Notice that since no label information is used, this does not compromise the conditions of Theorem 3.1 and thus p-value validity.

[1] A full binary tree is a tree in which every node in the tree has either 0 or 2 children.

[2] A perfect depth-K binary tree is a binary tree in which all interior nodes have two children and all leaves have the same depth K.

4.3.2 Consistency via context tree weighting

Intuitively, the finer the partition the larger the chances of finding revealing cells with the largest difference $\theta_C - \theta_0$ possible, which should be easier to detect. On the other hand, Eq. (4.16) shows that there is a cost on the convergence rate that is proportional to the size of the partition.

For this reason, we are interested in distributions that optimize the number of parameters at the same time they optimize the distribution itself, i.e., *twice-universal* distributions.

CTW. In order to build a twice-universal distribution, we use a Bayesian mixture of the distributions $\hat{P}_n^{\text{Jeffreys}(P)}$ associated to each partition $P \in \mathcal{P}_T$ using as weight $w_{nat}(P) \equiv 2^{-2|P|+1}$, obtaining therefore

$$\hat{P}_n^{\text{CTW}(\mathcal{P}_T)}\left(l^n|z^n\right) \equiv \sum_{P\in\mathcal{P}_T} w_{nat}(P)\hat{P}_n^{\text{Jeffreys}(P)}\left(l^n|z^n\right). \tag{4.22}$$

This can be efficiently computed using the recursive algorithm called *Context Tree Weighting (CTW)* [WST95] (with the generalization proposed in [WST96]), where z gives the context and each node can be split in only one way (accordingly to T). This combination of spatial partitions and CTW has already been used in [KSB08] to build universal portfolios.

For any cell C partitioned into a set of cells $\{C_i\}$, we have the following recursive formula

$$\hat{P}_n^{\text{CTW}(C)}\left(l^n|z^n\right) \equiv \frac{1}{2}\hat{P}_n^{\text{Jeffreys}(C)}\left(l^n|z^n\right) + \frac{1}{2}\prod_i \hat{P}_n^{\text{CTW}(C_i)}\left(l^n|z^n\right), \tag{4.23}$$

and for a cell C which cannot be split (leaf node)

$$\hat{P}_n^{\text{CTW}(C)}\left(l^n|z^n\right) \equiv \hat{P}_n^{\text{Jeffreys}(C)}\left(l^n|z^n\right). \tag{4.24}$$

Let C_0 be the cell corresponding to the root node of a tree T, then it can be proved by induction [WST95, Lemma 2] that

$$\hat{P}_n^{\text{CTW}(\mathcal{P}_T)}\left(l^n|z^n\right) = \hat{P}_n^{\text{CTW}(C_0)}\left(l^n|z^n\right). \tag{4.25}$$

The next lemma shows that $\hat{P}_n^{\text{CTW}(\mathcal{P}_T)}$ is asymptotically as good as any distribution $\mathbb{P}_{\boldsymbol{\theta}_P}$ such that $P \in \mathcal{P}_T$, the convergence rate depending on the size of P.

Lemma. 4.3. *$\hat{P}_n^{\text{CTW}(\mathcal{P}_T)}$ is* pointwise twice-universal *for the class of distributions defined by $\mathcal{P}_T$ since it satisfies for every l^n, z^n and for any partition $P \in \mathcal{P}_T$:*

$$-\frac{1}{n}\log \hat{P}_n^{\text{CTW}(\mathcal{P}_T)}\left(l^n|z^n\right) \leq -\frac{1}{n}\log \mathbb{P}_{\boldsymbol{\theta}_P}\left(L^n|Z^n\right) + O\left(|P|\frac{\log n}{2n}\right) \xrightarrow{n\to\infty} H_{\boldsymbol{\theta}_P}\left(L|Z\right) \ \ a.s.. \tag{4.26}$$

Proof. Follows from $\sum_{P\in\mathcal{P}_T} w_{nat}(P)\hat{P}_n^{\text{Jeffreys}(P)}\left(l^n|z^n\right) \geq w_{nat}(P)\hat{P}_n^{\text{Jeffreys}(P)}\left(l^n|z^n\right)$ for any $P \in \mathcal{P}_T$ and Lemma 4.2. □

Corollary. 4.2. *A test based on $\hat{P}_n^{\text{CTW}(\mathcal{P}_T)}$ is consistent if $\mathcal{P}_T$ contains a revealing partition.*

Proof. Follows from Lemma 4.3 and Theorem 4.1. □

Remark 4.2. *Since the root node of the hierarchical partition corresponds to $\mathbb{R}^d$, it is not necessary to use Jeffreys' estimation at this level and $\mathbb{P}_{\theta_0}\left(l^n\right) = \theta_0^{n_0}(1-\theta_0)^{n_1}$ can be used to avoid estimating θ_0 which is already known.*

Remark 4.3 (Sequential computation of CTW)**.** *At each internal node except the root (cf. Remark 4.2), by Eq. (4.23), CTW performs a Bayesian mixture of two models (Jeffreys and CTW from child nodes). This can be sequentially computed as in Eq. (3.38) and (3.39). Therefore, sequential computation of CTW only requires maintaining these weights for each internal node plus the counters mentioned in Remark 4.1. More precisely when a new observation (l, z) is processed, the tree is traversed from the root to a leaf node, following the path corresponding to the sequence of nested cells containing z, whose length is $O\left(\log N\right)$, with N the number of unlabeled samples.*

4.3.3 Increasing power using switch distributions

Since CTW is based on Bayesian mixtures it is also prone to the catch-up phenomenon (see Section 3.4). In order to avoid it and hopefully improve power, we use the recursive switch distribution called *Context Tree Switching* (CTS) proposed in [VNHB12]. The intuitive idea is to allow using simpler models (i.e., smaller partitions) when few observations are available, and then switch to more complex models (i.e., finer partitions) when more data is available.

Like in CTW (see Eq. (4.23)), the recursive formula considers at each internal node two possible models, i.e., the one given by Jeffreys' estimation for the corresponding cell and the one combining the models corresponding to each child node. These two models are indexed by the symbols a, b respectively.

Let $w_C(\cdot)$ be a prior over model index sequences $i^{n(C)} \equiv i_1, \ldots, i_{n(C)} \in \{a,b\}^{n(C)}$ at cell C, recursively defined by

$$w_C(i^{n(C)}) \equiv \begin{cases} 1 \text{ if } n(C) = 0 \\ \frac{1}{2} \text{ if } n(C) = 1 \\ w_s(i^{n(C)-1}) \left((1-\alpha_n)\mathbb{1}[i_{n(C)} = i_{n(C)-1}] + \alpha_n \mathbb{1}[i_{n(C)} \neq i_{n(C)-1}]\right), \text{ otherwise} \end{cases} \tag{4.27}$$

where $\alpha_n \equiv \frac{1}{n}$ is the switch rate. The previous equation calls for two comments:

- the first line is needed, as cells void of points appear in the formula below, and
- the last line corresponds to the cases where one does not switch models, and switches models, respectively.

To define the formula replacing Eq. (4.23) in the context of CTS, we shall need:

- $C(l^n)$: the subsequence of l^n corresponding to the symbols that fell in C
- $C(l^n)_{1:k}$: the first k symbols of $C(l^n)$
- $t_C(k)$: the smallest integer implicitly defined by the following equation:

$$\text{length}\left(C\left(l^{t_C(k)}\right)\right) = k. \tag{4.28}$$

The following equation replaces Eq. (4.23) and defines the CTS mixture at each internal node C (assuming two child nodes C_1 and C_2):

$$\hat{P}_n^{\text{CTS}(C)}(l^n|z^n) \equiv \sum_{i^{n(C)} \in \{a,b\}^{n(C)}} w_C(i^{n(C)}) \prod_{k=1}^{n(C)} [\mathbb{1}[i_k = a] \frac{\hat{P}_n^{\text{Jeffreys}(C)}\left(C(l^n)_{1:k}\right)}{\hat{P}_n^{\text{Jeffreys}(C)}\left(C(l^n)_{1:k-1}\right)} + \tag{4.29}$$

$$\mathbb{1}[i_k = b] \frac{\hat{P}_{t_{C(k)}}^{\text{CTS}(C_1)}\left(l^{t_{C(k)}}|z^{t_{C(k)}}\right)}{\hat{P}_{t_{C(k)}-1}^{\text{CTS}(C_1)}\left(l^{t_{C(k)}-1}|z^{t_{C(k)}-1}\right)} \frac{\hat{P}_{t_{C(k)}}^{\text{CTS}(C_2)}\left(l^{t_{C(k)}}|z^{t_{C(k)}}\right)}{\hat{P}_{t_{C(k)}-1}^{\text{CTS}(C_2)}\left(l^{t_{C(k)}-1}|z^{t_{C(k)}-1}\right)}]. \tag{4.30}$$

As shown in [VNHB12], universality is preserved, therefore yielding consistent tests when revealing partitions are used.

Corollary. 4.3. *A test based on $\hat{P}_n^{\text{CTS}(\mathcal{P}_T)}$ is consistent if $\mathcal{P}_T$ contains a revealing partition.*

Proof. Follows from [VNHB12, Thm. 3] and Theorem 4.1. □

Remark 4.4 (Sequential computation of CTS)**.** *CTS can be sequentially computed with the same asymptotic time and space complexity as CTW using the update equations given in [VNHB12, Sec. 3.1].*

4.4 Using an ensemble of trees

When considering finite length performance, we can be unlucky and obtain a bad set of hierarchical partitions. Therefore, in order to reduce variance and, hopefully, to improve the probability of rejecting under the alternative hypothesis, we use an ensemble of J random trees. The corresponding partitions are denoted $\{\mathcal{P}_{T_1}, \ldots, \mathcal{P}_{T_J}\}$. Using them, we define the following mixture with the uniform prior $w_{unif}(J) \equiv \frac{1}{J}$:

$$\hat{P}_n^{CT*2(\mathcal{P}_{T_1},\ldots,\mathcal{P}_{T_J})}(l^n|z^n) \equiv \sum_{j=1..J} w_{unif}(J)\hat{P}_n^{CT*(\mathcal{P}_{T_j})}(l^n|z^n)\,. \tag{4.31}$$

where CT*2 denotes either CTW2 or CTS2 depending on which distribution CT* (CTW or CTS respectively) is used in each tree of the ensemble.

Lemma. 4.4. *$\hat{P}_n^{CT*2(\mathcal{P}_{T_1},\ldots,\mathcal{P}_{T_J})}$ is pointwise twice-universal for the class of distributions defined by $\{\mathcal{P}_{T_1}, \ldots, \mathcal{P}_{T_J}\}$ since it satisfies for every l^n, z^n and for any partition $P \in \mathcal{P}_{T_j}$:*

$$-\frac{1}{n}\log \hat{P}_n^{CT*2(\mathcal{P}_{T_1},\ldots,\mathcal{P}_{T_J})}(l^n|z^n) \leq -\frac{1}{n}\log \mathbb{P}_{\boldsymbol{\theta}_P}(L^n|Z^n) + O\left(|P|\frac{\log n}{2n}\right) \tag{4.32}$$

$$\xrightarrow{n\to\infty} H_{\boldsymbol{\theta}_P}(L|Z) \;\; a.s.. \tag{4.33}$$

Proof. Follows from $\hat{P}_n^{CT*2(\mathcal{P}_{T_1},\ldots,\mathcal{P}_{T_J})}(l^n|z^n) \geq w_{unif}(J)\hat{P}_n^{CT*(\mathcal{P}_{T_j})}(l^n|z^n)$ for any j and Lemma 4.3. □

Corollary. 4.4. *A test based on $\hat{P}_n^{CT*2(\mathcal{P}_{T_1},\ldots,\mathcal{P}_{T_J})}$ is consistent if $\{\mathcal{P}_{T_1}, \ldots, \mathcal{P}_{T_J}\}$ contains a revealing partition.*

Proof. Follows from Lemma 4.4 and Theorem 4.1. □

4.5 Experiments

4.5.1 Implementation

The random projection context trees are represented by a binary tree structure in which each node representing a cell C stores the counters $n_l(C), l \in \{0, 1\}$. Internal nodes also store a representation of the hyperplane used by the RP tree to split the cell into its two children. Therefore, when a new sample is processed the counters and the weights to be updated are those located in the path from the root node to the leaf node reached by going to the proper side at each internal node according to the side of the hyperplane the sample is located at.

Observation. 2. *Processing a new observation has time complexity $O(J \log N)$.*

Observation. 3. *Potentially, the counters could overflow, requiring an arbitrary-precision counter or, otherwise, approximate counting (see, e.g., [Fla85]).*

4.5.2 Choice of parameters

We used $J = 1000$ trees and the whole sequence (since in this experiments they are known in advance) as unlabeled training set.

4.5.3 Results

We tested on the same datasets considered in Section 3.6 and we compared against the results presented there. The results are presented in Tables 4.1, 4.2 and 4.3.

Overall, we observe an empirical Type I error under its design value of $\alpha = 0.05$, which is expected since the p-value is valid but not exact.

In all the cases, CTS2 exhibited a larger power than CTW2, therefore confirming the interest of switching to avoid the catch-up phenomenon.

Table 4.1 Gaussian data in dimension $d = 4$. The symbol $\mathtt{H}_0$ or $\mathtt{H}_1$ at the beginning of each line indicates whether the null is true or false. Numbers indicate the percentage of trials for which the null hypothesis was rejected, given $\alpha = 0.05$. A total of 150 trials were done.

Case	$\mathrm{SW}_{\pi_U}^{(OS)}$	$\mathrm{SW}_{\pi_S}^{(OS)}$	$\mathrm{CTW2}^{(OS)}$	$\mathrm{CTS2}^{(OS)}$	TRank	$\mathrm{TRank}_{\mathrm{ref}}$	MMD_b
$\mathtt{H}_0$, Ex1	2	6	2.7	3.3	4.7	1	3.3
$\mathtt{H}_1$, Ex2	100	100	100	100	100	99	100
$\mathtt{H}_1$, Ex3a	20	9.3	32	61.3	40.7	45	90
$\mathtt{H}_1$, Ex3b	65.3	21.3	84.7	97.3	76	73	98.7

Table 4.2 Gaussian data in dimension $d = 10, 30$. For conventions, see the caption of Table 4.1. In this experiment, the data under $\mathtt{H}_0$ was generated by sampling from the mixture of both distributions. 100 trials were done. (NB: the conditions associated with $\mathrm{TRank}_{\mathrm{ref}}$ are specified in M. Depecker's PhD Thesis [Dep10].)

Case	$\mathrm{SW}_{\pi_U}^{(OS)}$	$\mathrm{SW}_{\pi_S}^{(OS)}$	$\mathrm{CTW2}^{(OS)}$	$\mathrm{CTS2}^{(OS)}$	TRank	$\mathrm{TRank}_{\mathrm{ref}}$	MMD_b
$\mathtt{H}_0, d = 10$	1	5	1	2	5		6
$\mathtt{H}_0, d = 30$	0	3	1	2	6		6
$\mathtt{H}_1, d = 10$	68	35	16	47	36	90	99
$\mathtt{H}_1, d = 30$	29.5	10	2	13	25	85	95

In the case Ex3, we observe a good performance of CTS2 that is comparable to MMDb, especially, when the sample size increases (Ex3b).

With respect to the higher dimensional cases of Table 4.2, we observe that CT*2 is prone to the curse of dimensionality as our KNN based methods, which is expected since they are based on spatial partitions.

In the subtle cases B2 and B3 of Table 4.3, we observe the ability of our method to automatically adapt to the right scale without requiring any scale related parameter to be set as it is the case for our KNN based mixtures and the Kernel optimized MMD versions.

4.6 Conclusions

This work introduces a sequential nonparametric two-sample test, based on sequential prediction of labels defining the two populations, with the novel characteristic of being semi-supervised.

We introduce the notion of revealing partition and use it to establish the consistency of the two-sample test. Our test is shown to be consistent when prediction is carried out using Context Tree Weighting and Context Tree Switching over a set of partitions containing a revealing one. These constructions do not

Table 4.3 Lattice of Gaussians in dimension $d = 2$. For conventions, see the caption of Table 4.1. The data under $\mathtt{H}_0$ was generated by from the mixture of both distributions. A total of 200 trials were done.

Case	$\mathrm{SW}_{\pi_U}^{(OS)}$	$\mathrm{SW}_{\pi_S}^{(OS)}$	$\mathrm{CTW2}^{(OS)}$	$\mathrm{CTS2}^{(OS)}$
$\mathtt{H}_0$, B1	0.5	5.5	2.5	2.5
$\mathtt{H}_1$, B2	27.5	16.5	100	100
$\mathtt{H}_1$, B3	96.5	91.5	100	100

Case	TRank	MMD_b	OPT	MxRt	MxMMD	L2	xvalc	med
$\mathtt{H}_0$, B1	6	5	4	2.5	2.5	4	7	7
$\mathtt{H}_1$, B2	100	6	18.5	15.5	15	14.5	15.5	5
$\mathtt{H}_1$, B3	100	6.5	21	18.5	15	15	21	7.5

require any parameter to choose in order to define the scale at which differences occur. Implementation-wise, the random spatial partitions are constructed using random projection trees. The time complexity of processing a new observation only depends on the number of unlabeled samples used to build the trees initially, which therefore remains constant. This property together with a negligible footprint per observation makes the test suitable for streaming data.

For future work, the problem of analyzing the probability of finding a revealing partition as a function of the number of trees used in the Bayesian mixtures is open. Also, accommodating data in a metric (non Euclidean) space, using e.g. metric trees, would widen the application spectrum of the method.

Chapter 5

Conclusion

5.1 Conclusions

This thesis was motivated by the idea of giving new insights and proposing new methods for two-sample comparison.

Although the natural framework was the one of statistical hypothesis testing, we did not want to propose "yet another two-sample test" since there are many of them (none uniformly better than the others) and it is relatively easy to invent new ones.

An important concern was the practical relevance of nonparametric multivariate tests. If a consistent test is used, any small difference is detected with probability arbitrarily close to 1, as long as enough samples are given. If data comes from random sources that are not identical in nature, it is natural to believe that slightly small differences actually exist and the test turns out to answer the question: "do we have enough samples so as to show that the sets come from different distributions?" which might not be of so much interest.

This straightforwardly led us to the problem of measuring an effect size which was an almost unexplored field for nonparametric multivariate two-sample tests. Using a novel combination of statistical learning, Morse theory and topological persistence, we made a progress in this direction by proposing a simplified decomposition of the Jensen-Shannon divergence, allowing the user to a have synthetic view, in the form of a bar plot, of the structure of the difference, even in high dimensions.

As we explored the literature, we found out how related dissimilarity measures, classification and two-sample testing are. This motivated the idea of building a two-sample test based on sequential soft classification, giving birth to the first nonparametric multivariate two-sample test that processes samples in a sequential manner and enjoys optional stopping, in contrast to classical two-sample testing that uses batch data.

In order to build sequential soft classifiers, we took two different directions. In the first one, we established an interesting connection between strongly pointwise consistent regressors and consistent tests. In the second one, we combined two powerful constructions, namely random projection trees and universal distributions based on context trees, to build sequential classifiers that exhibit automatic adaption to scale and yield powerful two-sample tests.

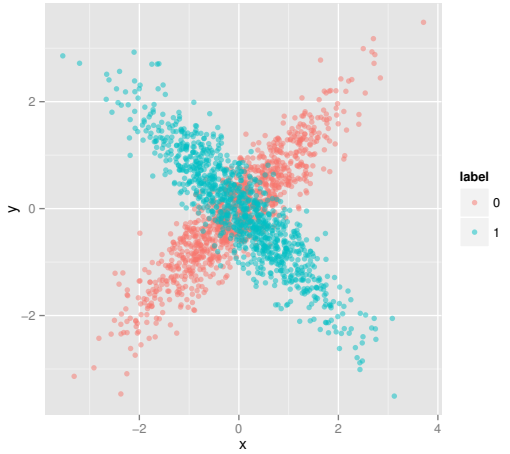
5.2 Discussion and Future Work

In many practical experiments, we observed that differences can often be spotted and more easily interpreted by univariate tests applied to each variable individually (in the same lines as the idea of one random projection being enough to detect any difference—see Section 1.4.2 and [CAFR07]).

For example, we compared two protein complexes datasets based on 19 energy measures[1]: one set consisting of complexes found in nature and the other one consisting of complexes produced by docking

[1] provided by Rosetta software, see `http://rosettadesign.med.unc.edu/documentation.php`.

Figure 5.1 Projection onto x or y axis does not allow to distinguish the datasets.



algorithms from their individual components. In this case, the univariate Mann-Whitney test and the Hodges-Lehmann estimator where able to show strong differences for some of the energies.

Another example consisted in comparing two datasets measuring cytokines' concentration[2] in sick (rheumatoid arthritis) and healthy individuals [DFS+15]. Here also, univariate analysis was able to detect and measure important differences in some individual cytokines' concentration.

Nevertheless, situations like the one depicted in Figure 5.1 are not detected with variable-wise testing. Therefore, we believe that, on a general perspective, the ability to provide nonparametric multivariate feedback (i.e., effect size) downstream two-sample testing may prove of ubiquitous interest in exploratory statistics and data science.

In order to continue this line of research, it would be interesting to characterize and provide theoretical guarantees on the cluster decomposition found by our procedure. Furthermore, it would be interesting to generalize the tool for data in (non-Euclidean) metric spaces in order to increase its range of applicability.

With respect to our sequential nonparametric two-sample tests, we believe that their novel features, namely sequentiality, suitability for streaming data and semi-supervised nature, clearly extend the range of applicability of classical two-sample tests, for example, towards big data streaming scenarios.

From a theoretical perspective, it would be interesting to find soft classifiers for data in (non-Euclidean) metric spaces that would provide consistency guarantees as well.

Another interesting line of research would be on the practical applied side where consistency might be of less importance: since any online soft classifier can be used to obtain a valid p-value, it is possible to use this framework to compare data as diverse as, for example, two streams of text messages.

[2]Cytokines are a broad category of small proteins that are involved in cell signaling.

Bibliography

[Alg92] P. Algoet. Universal schemes for prediction, gambling and portfolio selection. *The Annals of Probability*, 20(2):901–941, 1992.

[AS66] Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 131–142, 1966.

[Ass83] Patrice Assouad. Plongements lipschitziens dans $\mathbf{R}^n$. *Bull. Soc. Math. France*, 111(4):429–448, 1983.

[AZ05] B Aslan and G Zech. New test for the multivariate two-sample problem based on the concept of minimum energy. *Journal of Statistical Computation and Simulation*, 75(2):109–119, 2005.

[Bel61] Richard Ernest Bellman. *Adaptive control processes: a guided tour*, volume 4. Princeton university press Princeton, 1961.

[BF04] L. Baringhaus and C. Franz. On a new multivariate two-sample test. *Journal of Multivariate Analysis*, 88(1):190–206, 2004.

[BG05] G. Biau and L. Györfi. On the asymptotic properties of a nonparametric l1-test statistic of homogeneity. *Information Theory, IEEE Transactions on*, 51(11):3965–3973, 2005.

[BH04] A. Banyaga and D. Hurtubise. *Lectures on Morse Homology*. Kluwer, 2004.

[Bic69] P.J. Bickel. A distribution free version of the smirnov two sample test in the p-variate case. *The Annals of Mathematical Statistics*, 40(1):1–23, 1969.

[Bil13] P. Billingsley. *Convergence of probability measures (2nd Edition)*. John Wiley & Sons, 2013.

[Bon36] Carlo E Bonferroni. *Teoria statistica delle classi e calcolo delle probabilita*. Libreria internazionale Seeber, 1936.

[Bre57] L. Breiman. The individual ergodic theorem of information theory. *The Annals of Mathematical Statistics*, pages 809–811, 1957.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Bub15] P. Bubenik. Statistical topological data analysis using persistence landscapes. *J. of Machine Learning Research*, 16:77–102, 2015.

[BY01] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):pp. 1165–1188, 2001.

[CadOS11] F. Chazal, L.J. Guibas an dS.Y. Oudot, and P. Skraba. Persistence-based clustering in Riemannian manifolds. In *ACM SoCG*, pages 97–106, 2011. tomato-11.

[CAFR07] J.A. Cuesta-Albertos, R. Fraiman, and T. Ransford. A sharp form of the cramer–wold theorem. *Journal of Theoretical Probability*, 20(2):201–209, 2007.

[CB01] G. Casella and R. Berger. *Statistical inference*. Duxbury Press, 2001.

[CB06] Nicolo Cesa-Bianchi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[CCS11] F. Cazals and D. Cohen-Steiner. Reconstructing 3D compact sets. *Computational Geometry Theory and Applications*, 45(1-2):1–13, 2011.

[CD14] Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 3437–3445, 2014.

[CD15] F. Cazals and T. Dreyfus. SBL, the Structural Bioinformatics Library, 2015. http://sbl.inria.fr.

[CDV09] S. Clémençon, M. Depecker, and N. Vayatis. AUC optimization and the two-sample problem. *Advances in Neural Information Processing Systems*, 22:360–368, 2009.

[cga] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[CM02] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[Csi63] Imre Csiszár. Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der ergodizitat von markoffschen ketten. *Magyar. Tud. Akad. Mat. Kutató Int. Közl*, 8:85–108, 1963.

[CSZ+06] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. Semi-supervised learning. 2006.

[CT06] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley & Sons, 2006.

[Daw84] A Philip Dawid. Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society. Series A (General)*, pages 278–292, 1984.

[Dep10] Marine Depecker. *Méthodes d'apprentissage statistique pour le scoring*. PhD thesis, Télécom ParisTech, 2010.

[Dev87] Luc Devroye. *A course in density estimation*. Birkhauser Boston Inc., 1987.

[DF08] S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 537–546. ACM, 2008.

[DFS+15] C. Daridon, S. Fleischer, P. Shen, S. Ries, A. Lhéritier, F. Cazals, S. Fillatreau, and T. Dörner. Cytokine production by b cells in autoimmunity. *In Preparation*, 2015.

[DGL96] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Verlag, 1996.

[DMFN13] Misha Denil, David Matheson, and De Freitas Nando. Consistency of online random forests. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1256–1264, 2013.

[Dwa57] M. Dwass. Modified randomization tests for nonparametric hypotheses. *The Annals of Mathematical Statistics*, 28(1):181–187, 1957.

[DX13] H. Ding and J. Xu. FPTAS for minimizing earth mover's distance under rigid transformations. In *Algorithms–ESA 2013*, pages 397–408. Springer, 2013.

[E+04] Michael D Ernst et al. Permutation methods: a basis for exact inference. *Statistical Science*, 19(4):676–685, 2004.

[E+10] Tim Adriaan Lambertus van Erven et al. *When data compression and statistics disagree: two frequentist challenges for the minimum description length principle*. PhD thesis, Mathematical Institute, Faculty of Science, Leiden University, 2010.

[EH10] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. AMS, 2010.

[ES03] D.M. Endres and J.E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information theory*, 2003.

[FF87] G. Fasano and A. Franceschini. A multidimensional version of the kolmogorov-smirnov test. *Monthly Notices of the Royal Astronomical Society*, 225:155–170, 1987.

[Fis22] Ronald A Fisher. On the interpretation of $\chi 2$ from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, pages 87–94, 1922.

[fJWWW+15] Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, and Luca Scrucca. *caret: Classification and Regression Training*, 2015. R package version 6.0-41.

[FK97] A.T. Fomenko and T.L. Kunii. *Topological Modeling for visualization*. Springer, 1997.

[Fla85] Philippe Flajolet. Approximate counting: a detailed analysis. *BIT Numerical Mathematics*, 25(1):113–134, 1985.

[FLLRB12] Magalie Fromont, Béatrice Laurent, Matthieu Lerasle, and Patricia Reynaud-Bouret. Kernels based tests with non-asymptotic bootstrap approaches for two-sample problem. In *JMLR: Workshop and Conference Proceedings*, volume 23, pages 23–1, 2012.

[FM53] R. Fortet and E. Mourier. Convergence de la réparation empirique vers la réparation théorique. *Ann. Scient. Ecole Norm. Sup.*, 70(3):267–285, 1953.

[FP09] MP Fay and MA Proschan. Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics surveys*, 4:1–39, 2009.

[FR79] J. Friedman and L.C. Rafsky. Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *The Annals of Statistics*, 7(4):pp. 697–717, 1979.

[FRD11] R. Filipovych, S.M. Resnick, and C. Davatzikos. Semi-supervised cluster analysis of imaging data. *NeuroImage*, 54(3):2185–2197, 2011.

[Fri04] J. Friedman. On multivariate goodness-of-fit and two-sample testing. *Proceedings of Phystat2003, http://www. slac. stanford. edu/econf C*, 30908, 2004.

[FST73] Jerome H Friedman, Sam Steppel, and JW Tukey. *A nonparametric procedure for comparing multivariate point sets*. Number 153. Stanford Linear Accelerator Center Computation Research Group Technical Memo, 1973.

[GBR+06] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2006.

[GBR+12] A. Gretton, K.M. Borgwardt, J.R. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[Ges70] M.P. Gessaman. A consistent nonparametric multivariate density estimator based on statistically equivalent blocks. *The Annals of Mathematical Statistics*, 41(4):1344–1346, 1970.

[GFHS09] Arthur Gretton, Kenji Fukumizu, Zaid Harchaoui, and Bharath K Sriperumbudur. A fast, consistent kernel two-sample test. In *Advances in neural information processing systems*, pages 673–681, 2009.

[GK02] L. Györfi and A. Krzyzak. *A distribution-free theory of nonparametric regression*. Springer, 2002.

[GO80] Louis Gordon and Richard A Olshen. Consistent nonparametric regression from recursive partitioning schemes. *Journal of Multivariate Analysis*, 10(4):611–627, 1980.

[GO84] Louis Gordon and Richard A Olshen. Almost surely consistent nonparametric regression from recursive partitioning schemes. *Journal of Multivariate Analysis*, 15(2):147–163, 1984.

[Goo05] Phillip I Good. *Permutation, parametric and bootstrap tests of hypotheses*. Springer Science+ Business Media, 2005.

[Grü07] P.D. Grünwald. *The minimum description length principle*. MIT press, 2007.

[GSS+12] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B.K. Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems*, pages 1205–1213, 2012.

[Gyo81] L Gyorfi. The rate of convergence of-nn regression estimates and classification rules (corresp.). *IEEE Transactions on Information Theory*, 27(3):362–364, 1981.

[Hen88] N. Henze. A multivariate two-sample test based on the number of nearest neighbor type coincidences. *The Annals of Statistics*, pages 772–783, 1988.

[HJL63] Joseph L Hodges Jr and Erich L Lehmann. Estimates of location based on rank tests. *The Annals of Mathematical Statistics*, pages 598–611, 1963.

[HP99] Norbert Henze and Mathew D. Penrose. On the multivariate runs test. *The Annals of Statistics*, 27(1):pp. 290–298, 1999.

[HT02] P. Hall and N. Tajvidi. Permutation tests for equality of distributions in high-dimensional settings. *Biometrika*, 89(2):359, 2002.

[Jef46] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

[Joc86] Karl-Heinz Jockel. Finite sample properties and asymptotic efficiency of monte carlo tests. *The annals of Statistics*, pages 336–347, 1986.

[KD12] S. Kpotufe and S. Dasgupta. A tree-based regressor that adapts to intrinsic dimension. *J. of Computer and System Sciences*, 78(5):1496–1515, 2012.

[KKK62] S. Kullback, M. Kupperman, and H. H. Ku. Tests for contingency tables and markov chains. *Technometrics*, 4(4):pp. 573–608, 1962.

[Kol33] AN Kolmogorov. Sulla determinazione empirica delle leggi di probabilita. *Giorn. Ist. Ital. Attuari*, 4:1–11, 1933.

[Kpo11] Samory Kpotufe. k-nn regression adapts to local intrinsic dimension. In *Advances in Neural Information Processing Systems*, pages 729–737, 2011.

[KSB08] S.S. Kozat, A.C. Singer, and A.J. Bean. Universal portfolios via context trees. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 2093–2096. IEEE, 2008.

[KSHZ04] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004.

[KSW04] Jyrki Kivinen, Alexander J Smola, and Robert C Williamson. Online learning with kernels. *Signal Processing, IEEE Transactions on*, 52(8):2165–2176, 2004.

[KT81] R. Krichevsky and V. Trofimov. The performance of universal encoding. *Information Theory, IEEE Transactions on*, 27(2):199–207, 1981.

[Kul59] Solomon Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

[LC98] Y. LeCun and C. Cortes. The MNIST database of handwritten digits, 1998.

[Leh51] E. L. Lehmann. Consistency and unbiasedness of certain nonparametric tests. *The Annals of Mathematical Statistics*, 22(2):pp. 165–179, 1951.

[Lin91] J. Lin. Divergence measures based on the Shannon entropy. *Information Theory, IEEE Transactions on*, 37(1):145–151, 1991.

[LL04] Jun Li and Regina Y. Liu. New nonparametric tests of multivariate locations and scales using data depth. *Statistical Science*, 19(4):pp. 686–696, 2004.

[LPS99] Regina Y. Liu, Jesse M. Parelius, and Kesar Singh. Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *The Annals of Statistics*, 27(3):pp. 783–840, 1999.

[LR05] E Erich Leo Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer Science+ Business Media, 2005.

[LRH07] R.H.C. Lopes, I. Reid, and P.R. Hobson. The two-dimensional kolmogorov-smirnov test. In *XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research, Nikhef, Amsterdam, the Netherlands, April 23-27, 2007*, 2007.

[LV06] Friedrich Liese and Igor Vajda. On divergences and informations in statistics and information theory. *Information Theory, IEEE Transactions on*, 52(10):4394–4412, 2006.

[LY00] Lei Li and Bin Yu. Iterated logarithmic expansions of the pathwise code lengths for exponential families. *IEEE Transactions on Information Theory*, 46(7):2683–2689, 2000.

[LZ05] John Langford and Bianca Zadrozny. Estimating class membership probabilities using classifier learners. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 198–205, 2005.

[M+72] George Marsaglia et al. Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, 43(2):645–646, 1972.

[MB02] Grégoire Malandain and Jean-Daniel Boissonnat. Computing the diameter of a point set. *International Journal of Computational Geometry & Applications*, 12(6):489–510, December 2002.

[MCM12] V. Melnykov, W-C. Chen, and R. Maitra. MixSim: An R package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, 51(12):1–25, 2012.

[MF98] N. Merhav and M. Feder. Universal prediction. *Information Theory, IEEE Transactions on*, 44(6):2124–2147, 1998.

[MGS93] John W Miller, Rod Goodman, and Padhraic Smyth. On loss functions which minimize to conditional expected values and posterior probabilities. *Information Theory, IEEE Transactions on*, 39(4):1404–1408, 1993.

[Mil63] J.W. Milnor. *Morse Theory*. Princeton University Press, Princeton, NJ, 1963. m-mt-63.

[ML09] Marius Muja and David G Lowe. Flann, fast library for approximate nearest neighbors. In *International Conference on Computer Vision Theory and Applications (VISAPP'09)*, 2009.

[MO95] J. Möttönen and H. Oja. Multivariate spatial sign and rank methods. *Journal of Nonparametric Statistics*, 5(2):201–213, 1995.

[Mor63] Tetsuzo Morimoto. Markov processes and the h-theorem. *Journal of the Physical Society of Japan*, 18(3):328–331, 1963.

[MPB96] Jen-Fue Maa, Dennis K. Pearl, and Robert Bartoszynski. Reducing multidimensional two-sample data to one-dimensional interpoint comparisons. *The Annals of Statistics*, 24(3):pp. 1069–1074, 1996.

[Mul59] Mervin E Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2(4):19–20, 1959.

[MW47] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 18(1):50–60, 1947.

[Nob96] Andrew Nobel. Histogram regression estimation using data-dependent partitions. *The Annals of Statistics*, 24(3):1084–1105, 1996.

[NP33] J. Neyman and E. S. Pearson. On the Problem of the Most Efficient Tests of Statistical Hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933.

[Oja10] Hannu Oja. *Multivariate nonparametric methods with R: an approach based on spatial signs and ranks*. Springer Science & Business Media, 2010.

[OR04] H. Oja and R.H. Randles. Multivariate nonparametric tests. *Statistical Science*, 19(4):598–605, 2004.

[P+99] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[PC08] F. Pérez-Cruz. Kullback-Leibler divergence estimation of continuous distributions. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 1666–1670. Ieee, 2008.

[Pea83] JA Peacock. Two-dimensional goodness-of-fit testing in astronomy. *Monthly Notices of the Royal Astronomical Society*, 202:615–627, 1983.

[POSaH] G. Pau, A. Oles, M. Smith, and O. Sklyar andW. Huber. *EBImage: Image processing toolbox for R*. R package version 4.4.0.

[PS10] B. Phipson and G.K. Smyth. Permutation p-values should never be zero: calculating exact p-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1), 2010.

[Rab89] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Ris78] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[Ris87] Jorma Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 223–239, 1987.

[Ris96] J.J. Rissanen. Fisher information and stochastic complexity. *Information Theory, IEEE Transactions on*, 42(1):40–47, 1996.

[RM00] J. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Mathematical morphology*, page 187, 2000.

[Ros05] P.R. Rosenbaum. An exact distribution-free test comparing two multivariate distributions based on adjacency. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(4):515–530, 2005.

[RW11] Mark D Reid and Robert C Williamson. Information, divergence and risk for binary experiments. *The Journal of Machine Learning Research*, 12:731–817, 2011.

[Sch86] M.F. Schilling. Multivariate two-sample tests based on nearest neighbors. *Journal of the American Statistical Association*, 81(395):799–806, 1986.

[Ser06] Robert Serfling. Depth functions in nonparametric multivariate inference. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 72:1, 2006.

[SF12] Gail M Sullivan and Richard Feinn. Using effect size-or why the p value is not enough. *Journal of graduate medical education*, 4(3):279–282, 2012.

[Sht87] Y.M. Shtar'kov. Universal sequential coding of single messages. *Problemy Peredachi Informatsii*, 23(3):3–17, 1987.

[Sil86] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[SMFLPCAR08] Ricardo Santiago-Mozos, R Fernandez-Lorenzana, Fernando Perez-Cruz, and Antonio Artes-Rodriguez. On the uncertainty in sequential hypothesis testing. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 1223–1226. IEEE, 2008.

[Smi48] Nickolay Smirnov. Table for estimating the goodness of fit of empirical distributions. *The annals of mathematical statistics*, pages 279–281, 1948.

[SN07] J. Silva and S. Narayanan. Universal consistency of data-driven partitions for divergence estimation. In *IEEE International Symposium on Information Theory*, pages 2021–2025. Citeseer, 2007.

[SN10] Jorge Silva and Shrikanth S Narayanan. Information divergence estimation based on data-dependent partitions. *Journal of Statistical Planning and Inference*, 140(11):3180–3198, 2010.

[SRH+10] Sören Sonnenburg, Gunnar Rätsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, and Vojtěch Franc. The shogun machine learning toolbox. *The Journal of Machine Learning Research*, 11:1799–1802, 2010.

[SS02] B. Scholkopf and A.J. Smola. *Learning with kernels*. MIT press, 2002.

[SSVV11] G. Shafer, A. Shen, N. Vereshchagin, and V. Vovk. Test martingales, Bayes factors and p-values. *Statistical Science*, 26(1):84–101, 2011.

[Sto77] Charles J Stone. Consistent nonparametric regression. *The annals of statistics*, pages 595–620, 1977.

[Sto80] Charles J Stone. Optimal rates of convergence for nonparametric estimators. *The annals of Statistics*, pages 1348–1360, 1980.

[Sto82] Charles J Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, pages 1040–1053, 1982.

[STS09] L. Song, C.H. Teo, and A.J. Smola. Relative novelty detection. In *International Conference on Artificial Intelligence and Statistics*, pages 536–543, 2009.

[Tar83] R. E. Tarjan. *Data Structures and Network Algorithms*, volume 44 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[Vap99] V. Vapnik. *The nature of statistical learning theory*. springer, 1999.

[vdPG14] S. van der Pas and P. Grünwald. Almost the best of three worlds: Risk, consistency and optional stopping for the switch criterion in single parameter model selection. *arXiv preprint arXiv:1408.5724*, 2014.

[vEGdR12] T. van Erven, P. Grünwald, and S. de Rooij. Catching up faster by switching sooner: a predictive approach to adaptive estimation with an application to the aic–bic dilemma. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):361–417, 2012.

[VKD09] Nakul Verma, Samory Kpotufe, and Sanjoy Dasgupta. Which spatial partition trees are adaptive to intrinsic dimension? In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 565–574. AUAI Press, 2009.

[VNHB12] Joel Veness, Kee Siong Ng, Marcus Hutter, and Michael Bowling. Context tree switching. In *Data Compression Conference (DCC), 2012*, pages 327–336. IEEE, 2012.

[Wag07] E-J. Wagenmakers. A practical solution to the pervasive problems of p values. *Psychonomic bulletin & review*, 14(5):779–804, 2007.

[Wal45] Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[Wil91] David Williams. *Probability with martingales*. Cambridge university press, 1991.

[WKV05] Q. Wang, S.R. Kulkarni, and S. Verdú. Divergence estimation of continuous distributions based on data-dependent partitions. *Information Theory, IEEE Transactions on*, 51(9):3064–3074, 2005.

[WKV09] Q. Wang, S.R. Kulkarni, and S. Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *Information Theory, IEEE Transactions on*, 55(5):2392–2405, 2009.

[WST95] Frans M. J. Willems, Yuri M. Shtarkov, and Tjalling J. Tjalkens. The context-tree weighting method: Basic properties. *Information Theory, IEEE Transactions on*, 41(3):653–664, 1995.

[WST96] Frans M. J. Willems, Yuri M. Shtarkov, and Tjalling J. Tjalkens. Context weighting for general finite-context sources. *Information Theory, IEEE Transactions on*, 42(5):1514–1520, 1996.

[ZGB13] W. Zaremba, A. Gretton, and M. Blaschko. B-test: A non-parametric, low variance kernel two-sample test. In *Advances in Neural Information Processing Systems*, pages 755–763, 2013.

[ZS00] Yijun Zuo and Robert Serfling. General notions of statistical depth function. *The Annals of Statistics*, 28(2):pp. 461–482, 2000.

Appendix A

Resources: Algorithms and Software

A.1 Resources for Chapter 2

The two steps of our nonparametric effect size method are implemented using the following resources.

Exact and approximate nearest neighbors The first step consists in estimating a decomposition of the Jensen-Shannon divergence using k-nearest-neighbor based soft classification. Exact nearest neighbors soft classification is implemented in the `knn3` function in the R package `caret` [fJWWW+15]. Since this package does not allow incremental update of the training set, it is best suited for offline applications.

k-Nearest neighbor graph, watershed transform and topological persistence The second stage of our nonparametric effect size method combines a Morse theoretical analysis carried out on a k-nearest-neighbors-graph (connecting the data points, each equipped with its estimated discrepancy), with a simplification resorting to topological persistence (aiming at identifying stable features). These tools are implemented in the Structural Bioinformatics Library ([CD15] and `http://sbl.inria.fr`), and will be released in the fall of 2015.

A.2 Resources for Chapter 3

Our sequential two-sample test relies on k_n-nearest-neighbor online soft classification and it was compared against a number of tests.

Exact and approximate nearest neighbors For approximate and exact nearest neighbors in C++, one option is the Fast Library for Approximate Nearest Neighbors (FLANN)[ML09]. This library allows one to define a custom point type which can be equipped with a label. It is then easy to compute probability estimates by counting each type of label among the nearest neighbor set returned. Note that this implementation allows incremental updates of the training set.

Two-sample tests Useful implementations of two-sample tests were the following ones:

- MMD [GBR+12, GSS+12]: Matlab implementations of different MMD estimators and tests are provided by the authors[1]. Two versions of MMD are provided in the R package `kernlab` in the function `kmmd` [KSHZ04]. Several MMD related functions are provided in the C++ library SHOGUN [SRH+10], which has interfaces to Matlab, R, Octave and Python. It also includes a version of the linear time MMD estimator and an interface for processing streaming data (although sample size needs to be fixed in advance).

[1] Available at `www.gatsby.ucl.ac.uk/~gretton/mmd/mmd.htm` and `www.gatsby.ucl.ac.uk/~gretton/adaptMMD/adaptMMD.htm`.

- TreeRank (AUC based test) [CDV09]: `TreeRank::TwoSample`[2]
- Multivariate Cramer [BF04]: `cramer::cramer.test`
- Nearest-Neighbors [Sch86]: `MTSKNN::mtsknn`
- Mann-Whitney [MW47] (univariate): `stats::wilcox.test` and `coin::wilcox_test` (the latter provides exact p-values in the presence of ties).
- Kolmogorov-Smirnov [Kol33, Smi48] (univariate): `stats::ks.test`

A.3 Resources for Chapter 4

Our change detection test requires building hierarchical spatial partitions.

Space partitioning: Geometry The Computational Geometry Algorithms Library (CGAL) [cga] provides a large set of geometric operations, which are in particular useful for projecting, e.g,

```
CGAL::Vector_d::operator* (const Vector_d< Kernel > &w)
```

and for defining hyperplanes or other oriented hypersurfaces for partitioning, e.g.,

```
CGAL::Hyperplane_d
CGAL::Sphere_d
CGAL::Oriented_side
```

Random directions can be generated using the function

```
CGAL::Random_points_on_sphere_d<Point_d>(int dim, double r, Random &rnd=default_random)
```

or by generating independent gaussian random numbers for each component (see, e.g., [Mul59, M$^+$72]) using, e.g., the boost class[3]

```
boost::normal_distribution
```

Fast exact and approximate algorithms for diameter computation (required, for example, for RPTree-Mean version of Random Projection Trees [DF08]) were proposed in [MB02].[4]

Space partitioning: Hierarchy A hierarchical partitioning can be represented by a tree structure and an array containing all the sample points. Points do not need to be replicated in each node to represent the content of each cell. Instead, the array can be recursively sorted in a quicksort fashion by using as sorting criterion the side on the separating hypersurface on which the points lie.

Then, the elements contained in each cell can be simply represented by the boundaries delimiting the portion of the array that contains them.

[2] Available at `http://cran.r-project.org/src/contrib/Archive/TreeRank`
[3] Available at `www.boost.org`.
[4] A C implementation can be found at `http://www-sop.inria.fr/members/Gregoire.Malandain/diameter/`.

Nonparametric Methods for Learning and Detecting Multivariate Statistical Dissimilarity

In this thesis, we study problems related to learning and detecting multivariate statistical dissimilarity, which are of paramount importance for many statistical learning methods nowadays used in an increasingly number of fields. This thesis makes three contributions related to these problems.

The first contribution introduces a notion of multivariate nonparametric *effect size* shedding light on the nature of the dissimilarity detected between two datasets. Our two step method first decomposes a dissimilarity measure (Jensen-Shannon divergence) aiming at localizing the dissimilarity in the data embedding space, and then proceeds by aggregating points of high discrepancy and in spatial proximity into clusters.

The second contribution presents the first sequential nonparametric two-sample test. That is, instead of being given two sets of observations of fixed size, observations can be treated one at a time and, when strongly enough evidence has been found, the test can be stopped, yielding a more flexible procedure while keeping guaranteed type I error control. Additionally, under certain conditions,using nearest neighbors regression, when the number of observations tends to infinity, the test has a vanishing probability of type II error.

The third contribution presents a semi-supervised sequential nonparametric two-sample test, this time based on random spatial partitioning. The test also exhibits a vanishing type II error, under certain conditions on the partitions. This test automatically detects multi-scale differences. Processing a new observation has logarithmic time complexity w.r.t. the unlabeled training dataset size, and negligible memory footprint, which makes it suitable for streaming data.

Méthodes non-paramétriques pour l'apprentissage et la détection de dissimilarité statistique multivariée

Cette thèse présente trois contributions en lien avec l'apprentissage et la détection de dissimilarité statistique multivariée, problématique d'importance primordiale pour de nombreuses méthodes d'apprentissage utilisées dans un nombre croissant de domaines.

La première contribution introduit la notion de *taille d'effet* multivariée non-paramétrique, éclairant la nature de la dissimilarité détectée entre deux jeux de données, en deux étapes. La première consiste en une décomposition d'une mesure de dissimilarité (divergence de Jensen-Shannon) visant à la localiser dans l'espace ambiant, tandis que la seconde génère un résultat facilement interprétable en termes de grappes de points de forte discrépance et en proximité spatiale.

La seconde contribution présente le premier test non-paramétrique d'homogénéité séquentiel, traitant les données issues de deux jeux une à une–au lieu de considérer ceux-ci in extenso. Le test peut ainsi être arrêté dès qu'une évidence suffisamment forte est observée, offrant une flexibilité accrue tout en garantissant un contrôle de l'erreur de type I. En utilisant des régresseurs basés sur les plus proches voisins, sous certaines conditions, la convergence vers 0 de la probabilité d'erreur de type II est établie.

La troisième contribution présente un test non-paramétrique d'homogénéité séquentiel semi-supervisé, basé sur des partitions aléatoires de l'espace. Le test a aussi une probabilité d'erreur de type II tendant vers zéro sous certaines hypothèses. Il détecte automatiquement des différences multi-échelles. Le traitement d'une nouvelle observation requiert un coût logarithmique en la taille du jeu de données non étiquetées utilisé pour l'apprentissage, et une taille mémoire négligeable. Ceci le rend particulièrement adapté aux flux de données.